DISSERTATIONES
INFORMATICAE
UNIVERSITATIS
TARTUENSIS
66

MAHMOUD SHOUSH

Prescriptive Process Monitoring Under Uncertainty and Resource Constraints

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS

66

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS

66

MAHMOUD SHOUSH

Prescriptive Process Monitoring Under Uncertainty and Resource Constraints

UNIVERSITY OF TARTU
Press

Institute of Computer Science, Faculty of Science and Technology, University of Tartu, Estonia.

Dissertation has been accepted for the commencement of the degree of Doctor of Philosophy (PhD) in Computer Science on April 1, 2025 by the Council of the Institute of Computer Science, University of Tartu.

Supervisor

Prof.	Marlon Gerardo Dumas Menjivar University of Tartu, Estonia

Opponents

Prof.	Annalisa Appice University of Bari Aldo Moro, Italy
Assoc. Prof.	Sylvio Barbon Junior University of Trieste, Italy

The public defense will take place on May 29, 2025 at 14:15 in Narva Rd. 18-2048.

The publication of this dissertation was financed by the Institute of Computer Science, University of Tartu.

ISSN 2613-5906 (print) ISSN 2806-2345 (pdf)
ISBN 978-9916-27-868-0 (print) ISBN 978-9916-27-869-7 (pdf)

University of Tartu Press
`http://www.tyk.ee/`

ABSTRACT

In modern organizations, activities that are performed on a repeated basis are grouped into business processes. These business processes are continuously monitored and optimized to deliver value to clients and other stakeholders. Process mining is a set of techniques to support the analysis and optimization of business processes based on event logs generated during the execution of these processes. A subset of process mining techniques, known as predictive process monitoring, leverages these event logs to train machine learning models that predict future states or events of a business process. Predictive process monitoring techniques enable managers to see, for example, which executions of a business process are likely to lead to negative outcomes, such as customer complaints or lost sales opportunities. However, these techniques do not directly help managers determine what they can do to prevent these negative outcomes.

Prescriptive process monitoring is a related set of techniques that address this limitation by prescribing interventions that increase the probability of positive outcomes. For example, a predictive monitoring model may suggest that an applicant is likely to reject a loan offer. Meanwhile, a prescriptive monitoring approach may recommend that a loan officer (a *resource*) sends an additional offer (an *intervention*) to increase the probability that a loan contract is concluded (the *positive outcome*). The backbone of a prescriptive monitoring approach is an *intervention policy* — a decision procedure that determines which process instances (cases) require intervention, the state at which the intervention should be triggered, and the resource who should perform this intervention. The goal of a prescriptive process monitoring method is to maximize a *gain function*, which considers the benefits of preventing negative case outcomes and the cost of performing the interventions.

Existing approaches in the field of prescriptive process monitoring have two recurrent limitations. First, they do not take into account the uncertainty associated with the predictions they rely upon. Second, they assume that every intervention that they prescribe will be executed, overlooking the fact that each intervention requires time from a resource and, thus, only a limited number of interventions may be executed. In other words, existing methods in this field are not *uncertainty-aware* and *resource-aware*.

This thesis makes three research contributions that address these two limitations. The first contribution is a prescriptive process monitoring method that incorporates measures of uncertainty into intervention policies. An empirical evaluation of this method shows that the use of uncertainty measures improves the performance of the intervention policies, especially in the presence of tight resource constraints. The second and third contributions are both uncertainty-aware and resource-aware prescriptive process monitoring methods. The second contribution is a rule-based (white-box approach) to prescriptive process monitoring. This method allows users to specify intervention policies via filtering and ranking policies based on outputs produced by uncertainty-aware predictive, causal, and

survival models. The third contribution is a black-box prescriptive process monitoring method. This method uses online reinforcement learning to learn an intervention policy that combines inputs from predictive, causal, and survival models to maximize a gain function without requiring a user to specify any rules.

The methods proposed in this thesis are empirically evaluated using real-world datasets and compared to baseline methods that are not uncertainty-aware or resource-aware. The results demonstrate that intervention policies are more effective when they combine outputs of predictive, causal, and survival models in an uncertainty-aware manner and when they explicitly consider resource availability. All developed methods are released as open-source software, enabling other researchers to extend this work.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Acronyms

AUC Area Under the Curve.

CatBoost Categorical Boosting.
CATE Conditional Average Treatment Effects.
CI Causal Inference.
CP Conformal Prediction.

DL Deep Learning.

GBT Gradient Boosted Trees.

IP Intervention Policy.
ITE Individual Treatment Effects.

LightGBM Light Gradient Boosting Machine.

MAE Mean Absolute Error.
ML Machine Learning.

PM Process Mining.
PPM Predictive Process Monitoring.
PrPM Prescriptive Process Monitoring.

RCTs Randomized Controlled Trials.
RF Random Forest.
RL Reinforcement Learning.
RMSE Root Mean Squared Error.
ROC Receiver Operating Characteristic.
R^2 R-squared.

TU Total Uncertainty.

XGBoost Extreme Gradient Boosting.

LIST OF ORIGINAL PUBLICATIONS

Publications included in the thesis

[1] **Mahmoud Shoush** and Marlon Dumas. "White box specification of intervention policies for prescriptive process monitoring". In: *Data & Knowledge Engineering* 155 (2024), 102379.
DOI: `https://doi.org/10.1016/j.datak.2024.102379`.
Lead author. The author implemented and analyzed the experiments and contributed substantially to the ideas and the writing.

[2] **Mahmoud Shoush** and Marlon Dumas. "Prescriptive Process Monitoring Under Resource Constraints: A Reinforcement Learning Approach". In: *KI-Künstliche Intelligenz* (2024), 1–22.
DOI: `https://doi.org/10.1007/s13218-024-00881-6`.
Lead author. The author implemented and analyzed the experiments and contributed substantially to the ideas and the writing.

[3] **Mahmoud Shoush** and Marlon Dumas. "Intervening With Confidence: Conformal Prescriptive Monitoring of Business Processes". In: *International Workshop on Process Management in the AI era (PMAI23@IJCAI)*. Vol. 3569. 2023, 1–12.
URL: `https://ceur-ws.org/Vol-3569/paper1.pdf`.
Lead author. The author implemented and analyzed the experiments and contributed substantially to the ideas and the writing.

[4] **Mahmoud Shoush** and Marlon Dumas. "When to Intervene? Prescriptive Process Monitoring Under Uncertainty and Resource Constraints". In: *Business Process Management (BPM Forum)*. Vol. 458. Lecture Notes in Business Information Processing. Springer, 2022, 207–223.
DOI: `http://dx.doi.org/10.1007/978-3-031-16171-1_13`.
Lead author. The author implemented and analyzed the experiments and contributed substantially to the ideas and the writing.

[5] **Mahmoud Shoush** and Marlon Dumas. "Prescriptive Process Monitoring Under Resource Constraints: A Causal Inference Approach". In: *International Conference on Process Mining (ICPM Workshops)*. Vol. 433. Lecture Notes in Business Information Processing. Springer, 2021, 180–193.
DOI: `https://doi.org/10.1007/978-3-030-98581-3_14`.
Lead author. The author implemented and analyzed the experiments and contributed substantially to the ideas and the writing.

Publications out of the scope of the thesis

[1] Zahra Dasht Bozorgi, Marlon Dumas, Marcello La Rosa, Artem Polyvyanyy, **Mahmoud Shoush**, and Irene Teinemaa. "Learning When to Treat Business Processes: Prescriptive Process Monitoring with Causal

Inference and Reinforcement Learning”. In: *Advanced Information Systems Engineering (CAiSE)*. Vol. 13901. Lecture Notes in Computer Science. Springer, 2023, 364–380.
DOI: `https://doi.org/10.1007/978-3-031-34560-9_22`.

[2] Kateryna Kubrak, Lana Botchorishvili, Fredrik Milani, Marlon Dumas, Alexander Nolte, **Mahmoud Shoush**, and Zhaosi Qu. “Kairos: A Tool for Prescriptive Monitoring of Business Processes”. In: *International Conference on Process Mining (ICPM Demo)*. Vol. 3648. 2023. 1–5.
DOI: `https://ceur-ws.org/Vol-3648/paper_214.pdf`.

1. INTRODUCTION

To remain competitive, organizations need to continuously improve their operations to provide sustained business value to clients, employees, and other stakeholders. For instance, when it comes to customer-facing processes, organizations are on a constant quest to reduce response times, costs, and error rates [185]. To achieve these objectives, larger organizations group together their activities into well-defined business processes that are continuously monitored, analyzed, and redesigned [4].

A *business process* is "a collection of interrelated events, activities, and decision points that involve multiple actors and objects, collectively leading to an outcome that is of value to at least one customer" [4]. These processes are typically supported by process-aware information systems that track and log their activities in artifacts known as *event logs*. In a nutshell, an event log is a collection of records, such that each record describes a state change of an activity instance (e.g., the start of an activity instance, the end, or both), alongside data attributes collected as part of this state change [5].

Process mining is a collection of techniques that leverage event logs to analyze, monitor, and identify opportunities for enhancing business process performance. Process mining enables organizations to gain insights into the causes of bottlenecks, to evaluate the effectiveness of business rules, and to visualize performance improvements, among other goals.

Predictive Process Monitoring (PPM) is a set of techniques that combine process mining and machine learning methods to predict future states or future events during the execution of business processes [187, 188, 189, 190, 191, 192, 18, 10]. For example, in a loan origination process, a predictive process monitoring tool can identify applicants likely to decline a loan offer based on their profiles and the steps they have undergone with the bank. While these predictions are valuable, their practical utility is limited if organizations cannot influence the predicted outcomes. Knowing that an applicant is likely to decline an offer is useful, but without the means to intervene and alter this outcome, these predictive insights can only have a limited impact.

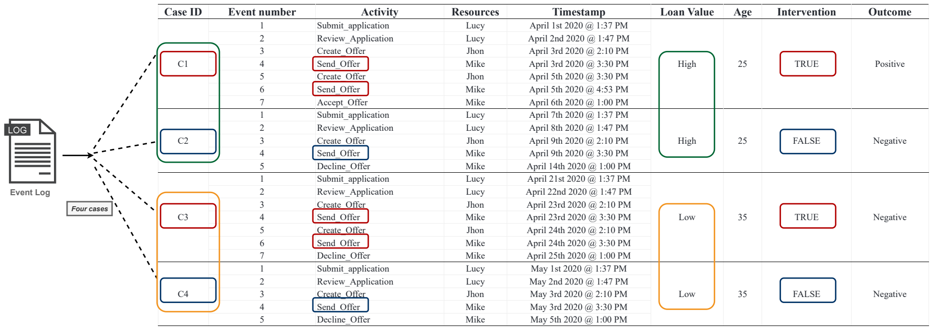
The emerging field of *prescriptive process monitoring* aims to translate predictions about future states or events in a process, into actions that have a positive impact on the performance of the process [45]. This thesis seeks to advance the state-of-the-art in the field of prescriptive process monitoring.

1.1. Problem Statement

Prescriptive Process Monitoring (PrPM) is a set of techniques to optimize the execution of business processes by prescribing or recommending specific actions (herein called *interventions*) during the execution of a business process instance (herein called a *case*) [194, 45]. Unlike predictive process monitoring, which focuses on forecasting future outcomes, PrPM takes a step further by proactively influencing these outcomes by recommending or triggering runtime interventions [193]. The aim of these interventions is to improve the process along one or more performance objectives, such as increasing the *success rate* of the process, i.e., the proportion of cases that end in a desired (positive) outcome.

At the core of a PrPM approach is the concept of learning an *Intervention Policy (IP)* from historical data. An intervention policy is a decision procedure that determines which cases require intervention, when the intervention should be triggered, and which resource (e.g., a process worker) should carry it out. The historical data used to develop this policy typically takes the form of an *event log*, that captures the execution history of activities across multiple cases within a process.

For example, consider a loan origination process, as depicted in Fig. 1. An event log in this context may consist of four cases: $C1$, $C2$, $C3$, and $C4$. A case begins when a loan application is submitted and concludes with either a positive outcome, where the applicant accepts the loan offer (e.g., $C1$), or a negative outcome, where the offer is declined (e.g., $C2$). So, to this end, a predictive monitoring model trained on these cases can be deployed at runtime to estimate the likelihood of each case's outcome at various stages. Suppose the model predicts a high probability that an applicant will reject the offer. In that case, the PrPM approach may recommend an intervention, such as having a loan officer (the resource) send a second offer (the intervention) to increase the chances of acceptance and, consequently, a positive outcome. In this scenario, case $C1$ receives an additional offer, resulting in a positive outcome, whereas case $C2$ receives no intervention, leading to a negative outcome.



Case ID	Event number	Activity	Resources	Timestamp	Loan Value	Age	Intervention	Outcome
C1	1	Submit_application	Lucy	April 1st 2020 @ 1:37 PM	High	25	TRUE	Positive
	2	Review_Application	Lucy	April 2nd 2020 @ 1:47 PM				
	3	Create_Offer	Jhon	April 3rd 2020 @ 2:10 PM				
	4	Send_Offer	Mike	April 3rd 2020 @ 3:30 PM				
	5	Create_Offer	Jhon	April 5th 2020 @ 3:30 PM				
	6	Send_Offer	Mike	April 5th 2020 @ 4:53 PM				
	7	Accept_Offer	Mike	April 6th 2020 @ 1:00 PM				
C2	1	Submit_application	Lucy	April 7th 2020 @ 1:37 PM	High	25	FALSE	Negative
	2	Review_Application	Lucy	April 8th 2020 @ 1:47 PM				
	3	Create_Offer	Jhon	April 9th 2020 @ 2:10 PM				
	4	Send_Offer	Mike	April 9th 2020 @ 3:30 PM				
	5	Decline_Offer	Mike	April 14th 2020 @ 1:00 PM				
C3	1	Submit_application	Lucy	April 21st 2020 @ 1:37 PM	Low	35	TRUE	Negative
	2	Review_Application	Lucy	April 22nd 2020 @ 1:47 PM				
	3	Create_Offer	Jhon	April 23rd 2020 @ 2:10 PM				
	4	Send_Offer	Mike	April 23rd 2020 @ 3:30 PM				
	5	Create_Offer	Jhon	April 24th 2020 @ 2:10 PM				
	6	Send_Offer	Mike	April 24th 2020 @ 3:30 PM				
	7	Decline_Offer	Mike	April 25th 2020 @ 1:00 PM				
C4	1	Submit_application	Lucy	May 1st 2020 @ 1:37 PM	Low	35	FALSE	Negative
	2	Review_Application	Lucy	May 2nd 2020 @ 1:47 PM				
	3	Create_Offer	Jhon	May 3rd 2020 @ 2:10 PM				
	4	Send_Offer	Mike	May 3rd 2020 @ 3:30 PM				
	5	Decline_Offer	Mike	May 5th 2020 @ 1:00 PM				

Figure 1. Event log with four cases.

However, suppose an intervention policy is developed based on a small sample, such as the four cases in Fig. 1. In this scenario, the policy may misleadingly suggest that interventions are highly effective for high-value loan cases like $C1$, where the intervention leads to a positive outcome 100% of the time. Meanwhile, the intervention has no effect on low-value cases such as $C3$ and $C4$. Despite this conclusion, the small sample size introduces significant uncertainty, making it difficult to generalize the policy. Applying such a policy in practice could lead to interventions without benefit, wasting time and effort, eventually resulting in a suboptimal policy. Uncertainty is inherent, to one extent or another, in any predictive model and needs to be taken into account when decisions are made on the basis of the outputs of such a model. Yet, existing PrPM approaches do no t account for uncertainty.

Moreover, existing PrPM approaches assume that interventions can be applied immediately and for all cases requiring intervention, without considering resource constraints. In practice, interventions require time and commitment from limited resources, such as a loan officer preparing a second offer. For instance, if ten ongoing cases in a loan origination process may benefit from an intervention, but only two loan officers are available, intervening in all cases may not be feasible. Failing to account for limited resource availability can lead to over-prescription of interventions, eventually reducing the effectiveness of the intervention policy.

This thesis addresses the problem of deriving intervention policies from event logs of business processes, in the context where the number of resources available to execute the interventions is limited and where the output of the predictive models used to decide which cases warrant an intervention have a certain level of uncertainty. In other words, the thesis addresses the problem of deriving intervention policies for prescriptive process monitoring that are uncertainty-aware and resource-aware.

1.2. Scope and Research Questions

The field of prescriptive process mining is still evolving and has gained increasing attention in recent years. This field encompasses various objectives and types of interventions, depending on the specific improvement goals. For instance, one prescriptive objective might aim to increase the rate of cases ending with a positive outcome, while another could focus on reducing cycle time. The prescribed intervention can also vary significantly, ranging from continuous recommendations, such as suggesting the next best activity throughout the process execution to improve a key performance indicator, to discrete interventions triggered to prevent a negative outcome. Moreover, multiple objectives and prescription types can be combined to address complex scenarios.

Given the broad scope of PrPM and the complexity involved in developing a fully functional prescriptive monitoring system for business processes, this thesis narrows its focus to a specific scenario. Specifically, we address scenarios where the prescribed action is one of the activities recorded in the event log and designated by a domain expert as a possible intervention. This intervention is intended to improve a particular objective, such as the rate of positive outcomes. Although real-world applications often involve multiple potential interventions within a single process case, this thesis simplifies the problem by considering a *binary inter-*

vention scenario. In this scenario, only a single intervention is considered, and the decision is whether or not to apply it.

Furthermore, this thesis assumes that the objective is to influence a *binary process outcome*—either a positive outcome, which generates value for the organization, or a negative outcome, which does not. This allows us to focus on improving decision-making processes regarding when and how to intervene during ongoing cases, enhancing the chances of achieving positive outcomes.

Although interventions can be beneficial, they often incur costs, such as direct monetary expenses. Despite these costs, applying an intervention remains advantageous if the benefit of achieving a positive outcome outweighs the associated expenses. However, these interventions also consume resources with limited capacity. Allocating scarce resources to unnecessary, uncertain, or costly interventions can be particularly problematic in a resource-constrained environment, leading to wasted efforts and resources. While a traditional prescriptive monitoring approach might focus solely on optimizing the primary objective, it is important to incorporate considerations of costs, uncertainties, and resource constraints to ensure that recommendations are both certain and resource-aware.

Accordingly, this thesis addresses the challenge of building a prescriptive monitoring system that not only optimizes process outcomes but also takes into account uncertainties, costs, and the limitations of available resources. Specifically, in this thesis, we focus on a particular instance of the general PrPM problem to answer the following overarching question:

> *Given an event log that contains completed cases of a process, where interventions have been applied to some cases, and functions that assign a cost to each intervention, how can we learn an intervention policy that recommends interventions during ongoing cases of this process in a way that maximizes a performance measure, i.e., the total gain, in a way that is uncertainty-aware and resource-constrained?*

In the presence of uncertainty and limited resources, one important aspect to consider when addressing this research question is identifying the intended users of the prescriptive system. According to the work done by Kubrak et al. [195], intended users can be categorized into three groups: process analysts, operational managers, and tactical managers. Process analysts are interested in analyzing historical process executions in an offline manner to identify mid-to-long-term improvements. On the other hand, operational managers work in a dynamic environment, dealing with ongoing process cases as they are executed at runtime, often requiring quick and informed decision-making. Tactical managers are concerned with optimizing the process in the long term, in particular by making sure that operational managers have an adequate number of resources and the right data and tools to coordinate the processes under their purview.

Additionally, based on the literature review (explored later), we observe that existing PrPM approaches can be divided into two broad streams: one stream relies on *white-box* decision procedures where analysts manually specify decision rules and thresholds. At the same time, the other uses a *black-box*, data-driven approach to derive intervention policies automatically. White-box approaches give more visibility and control to tactical process managers, by allowing them to understand the behavior of the intervention policies and to tune them according to their requirements. Meanwhile, data-driven policies are more adaptable, as they auto-tune themselves as more data becomes available. Black-box intervention policies are also expected to be more effective, since they are optimized to maximize the total gain that the intervention policy would have produced on historical observations.

This thesis explores methods for uncertainty-aware and resource-aware PrPM that address the needs of both operational and tactical managers. Accordingly, the methods explored in this thesis cover both white-box and black-box approaches.

Within this context, the thesis covers three areas of exploration:

- Uncertainty-Aware PrPM: Developing methods to incorporate uncertainty into the decision-making process for recommending operational interventions. The goal is to ensure that predictions are reliable and that decisions can be made confidently, even when uncertainty is present.
- White-Box PrPM: Creating transparent, rule-based policies that manage limited resources effectively. These policies aim to ensure that operational interventions are recommended and executed to optimize resource allocation while maximizing process performance, ensuring that operational interventions are interpretable.
- Black-Box PrPM: Developing data-driven, often indefinite policies that effectively manage limited resources. These policies aim to ensure that operational interventions are recommended and executed to optimize resource allocation and process performance that users cannot easily interpret.

Given the above three areas of exploration, we decompose the overarching research question of the thesis into three more specific research questions (RQs), as follows:

RQ1: *How can we develop intervention policies that effectively incorporate uncertainty into decision-making, ensuring that predictions are reliable and decisions are made confidently to maximize a total gain function?*

RQ2: *How can business stakeholders define transparent, rule-based white-box intervention policies that manage limited resources efficiently while optimizing the total gain of the resulting intervention policy in a resource-constrained setting?*

RQ3: *How can we design data-driven, black-box intervention policies that manage limited resources effectively, optimizing resource allocation and process performance, i.e., total gain, within a resource-constrained setting?*

1.3. Research Method

Given that this research falls within the field of information systems, the thesis follows the design science guidelines proposed by Hevner et al. [196] to address the research questions. The key aspects of the methodology are outlined below:

Design as an Artifact: The outcome of this Ph.D. project is the development of artifacts in the form of systems designed to trigger runtime interventions for business processes. These systems determine who requires intervention, when the intervention should be triggered, and who should execute it at a given state of a process case. The artifacts include models, algorithms, and decision-making frameworks that integrate the complexities of uncertainty and limited resources into intervention policies. Portions of these artifacts have been implemented in software through collaboration with colleagues, and these implementations are publicly available online. Links to this software are provided in the corresponding contribution chapters of this thesis.

Problem Relevance: This research addresses a critical problem in prescriptive process monitoring—how to make effective, uncertainty-aware, and resource-efficient intervention decisions during the execution of business processes. This problem has been recognized as relevant by both practitioners, who have expressed a need for effective systems to improve operational efficiency and process outcomes through qualitative interviews [195] and academics, as demonstrated by previous research on prescriptive process monitoring [45].

Design Evaluation: The developed artifacts are rigorously evaluated using real-life datasets with varying characteristics to assess their effectiveness. Evaluation methods include empirical analysis, which tests the systems' ability to make accurate and resource-efficient intervention decisions under uncertainty. The evaluation results provide insights into the performance of the proposed systems and their potential for real-world application.

Research Contribution: The research contributes to both academic and practitioner communities by advancing the field of prescriptive process monitoring and addressing a previously unsolved problem. It introduces new methods for handling uncertainty and resource constraints in decision-making processes, offering practical solutions that can be applied in various business contexts.

Research Rigor: The research is conducted with methodological rigor to ensure the reliability and validity of the findings. An extensive literature review, detailed in Chapter 3, is conducted to identify existing techniques and evaluation approaches in process mining, machine learning, and decision-making under uncertainty. State-of-the-art methods are then used to develop and test the artifacts, ensuring the research is grounded in established theories and practices.

Communication of Research: The results and contributions of this research are effectively communicated to both academic and practitioner audiences. This is achieved through publications in journals and conference proceedings, oral presentations, and the public release of a software tool.

1.4. Contributions

This thesis aims to develop and operationalize approaches for prescriptive process monitoring of business processes. These approaches, aligned with the defined research questions, aim to prescribe interventions throughout the execution of process cases until their completion, particularly in environments characterized by uncertainty and resource constraints. Consequently, this thesis makes three key contributions to the area of prescriptive analytics for business processes, as outlined below.

Contribution 1: Propose an Uncertainty-Aware Prescriptive Process Monitoring Approach (Chapter 4). This contribution extends existing PrPM work by developing an uncertainty-aware prescriptive process monitoring approach, termed *UN-PrPM*. The goal is to develop an intervention policy determining when to trigger interventions to maximize a total gain function without considering resource limitations. This approach triggers interventions for all cases as needed, irrespective of resource constraints. The process begins with discussing the predictive monitoring pipeline and explores methods for estimating uncertainties associated with predictions, focusing on uncertainty quantification and conformal prediction. The approach integrates these uncertainty estimations into the prescriptive model, resulting in a *UN-PrPM*. The effectiveness of this approach is evaluated using real-life event logs representing a loan origination process and is compared against a baseline relying solely on process outcome predictions.

This contribution, where we introduced the integration of uncertainty in PrPM, was initially published in the proceedings of the Business Process Management (BPM) Forum [315], where we discussed the application of uncertainty quantification in PrPM. The subsequent introduction of the conformal prediction method in this context was later presented and published in the proceedings of the 2nd International Workshop on Process Management in the AI Era [316].

Lead author. The author conceived the idea, designed the methodology, implemented the system, conducted the experiments, analyzed the results, and wrote the manuscript.

Contribution 2: Design of a Prescriptive Process Monitoring Approach Under Resource Constraints: A White-Box Approach (Chapter 5). This contribution presents a white-box prescriptive process monitoring approach, termed *WB-PrPM*, emphasizing transparency and interpretability in intervention policies. We emphasize that the white-box nature of an approach is determined by how the intervention policy is constructed. These policies are developed using parameterized factors derived from black-box techniques.

The *WB-PrPM* extends the *UN-PrPM* to be resource-aware and resource-constrained, allowing stakeholders to understand the rationale behind intervention recommendations. This transparency aids in balancing the necessity of effective interventions with the practical constraints of limited resources. The approach integrates machine learning techniques and allows stakeholders to specify intervention policies through filtering and ranking mechanisms that consider the importance, urgency, uncertainty, and available intervention capacity. The effectiveness of the *WB-PrPM* is empirically evaluated across various resource utilization levels, focusing on the total gain achieved compared to white-box baseline methods that do not consider resource constraints or uncertainty parameters.

Recalling the loan origination process in Fig. 1 illustrates the importance of intervention policies in PrPM. A white-box intervention policy enables process analysts to explicitly define rules for when and how interventions should be triggered. For example, suppose a predictive monitoring model estimates that an ongoing case (e.g., $C2$) has a 60% probability of rejection. In that case, an analyst can set a predefined rule to trigger an intervention—such as sending a second loan offer—only if this probability exceeds a chosen threshold (e.g., 70%). Conversely, for a more relaxed policy, the threshold can be adjusted to 50%, ensuring flexibility in decision-making. White-box approaches also allow analysts to incorporate uncertainty by setting conservative intervention thresholds when confidence in predictions is low.

An earlier version of this contribution was published in the proceedings of the 2nd International Workshop on Process Management in the AI Era [315]. Additionally, part of this work, which introduced the use of causal inference in business processes to evaluate the effectiveness of interventions on case outcomes, was presented at the Machine Learning for Process Mining (ML4PM) workshop at the International Conference on Process Mining [317]. The complete white-box method for PrPM was subsequently published in an article in Data and Knowledge Engineering [342].

Lead author. The author conceived the idea, designed the methodology, implemented the system, conducted the experiments, analyzed the results, and wrote the manuscript.

Contribution 3: Design of a Prescriptive Process Monitoring Approach Under Resource Constraints: A Black-Box Approach (Chapter 6). This contribution introduces a black-box approach to prescriptive process monitoring, termed *BB-PrPM*, which leverages data-driven methods to develop intervention policies at runtime. This approach serves as an alternative to *WB-PrPM*, mitigating its limitations by abstracting the complexity of policy formulation into a data-driven model. The *BB-PrPM* employs an online Reinforcement Learning (RL) approach that triggers interventions based on necessity, timeliness, effect, the uncertainty of predictions, and level of resource utilization. This approach aims to optimize intervention policies while minimizing human error and time spent on manual policy specification. The *BB-PrPM* is empirically evaluated for its convergence and performance metrics, with comparisons made to reinforcement learning-based baseline methods that do not account for resource constraints.

Unlike white-box approaches, which rely on predefined rules, black-box methods automate policy learning through RL. The system continuously refines intervention strategies based on past outcomes, adapting to changing conditions without direct input from process analysts. For instance, in the loan origination scenario, *BB-PrPM* learns whether intervening at a 60% rejection probability or sending a second loan offer improves acceptance rates, particularly in specific cases like high-value loans (e.g., $C1$ in Fig. 1). Over time, it optimizes interventions based on observed effectiveness. While this adaptability enhances efficiency, it reduces transparency, making it harder to explain why certain interventions were applied or omitted. The choice between white-box and black-box approaches ultimately depends on whether interpretability and control or automation and adaptability are the priority.

This contribution was published as an article in KI - Künstliche Intelligenz, the German Journal of Artificial Intelligence [343].

Lead author. The author conceived the idea, designed the methodology, implemented the system, conducted the experiments, analyzed the results, and wrote the manuscript.

1.5. Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 provides essential background, covering machine learning fundamentals, deep learning, process mining, and their intersection. It also introduces key concepts from causal inference, survival analysis, and reinforcement learning relevant to subsequent chapters.

Chapter 3 reviews the current state-of-the-art in prescriptive process monitoring for business processes. The chapter categorizes existing methods based on the various improvement objectives they aim to optimize, the types of interventions they make, the input data they require, and the types of intervention policy learning. We discuss the limitations of existing approaches and highlight the research gaps, particularly the need for integrating uncertainty and resource constraints into prescriptive monitoring.

Chapter 4 presents the first contribution of the thesis: an approach for incorporating uncertainty into prescriptive process monitoring. We explore uncertainty quantification and conformal prediction techniques and discuss how they can be integrated into prescriptive monitoring to develop uncertainty-aware intervention policy. Chapter 5 introduces the second key contribution, a white-box approach to prescriptive process monitoring that allows stakeholders to specify intervention policies under resource constraints. Chapter 6 presents the third contribution, a black-box prescriptive process monitoring approach that uses online reinforcement learning to manage interventions under resource constraints.

Finally, Chapter 7 summarizes the key findings and contributions of the thesis. It discusses the broader impact of the research on the field of prescriptive analytics and identifies potential areas for future research.

2. BACKGROUND

Prescriptive process monitoring (PrPM) is an interdisciplinary research area combining machine learning, statistical analysis, and process mining techniques to develop intervention policies to optimize business process performance. This section outlines the fundamental concepts from these fields, which are essential for understanding the subsequent chapters.

We begin by discussing key concepts in machine learning, focusing on supervised learning, uncertainty estimation, and reinforcement learning. Next, we explore two additional fields—causal inference and survival analysis—that are closely related to this research. Finally, we introduce process mining, with an emphasis on predictive process monitoring, a subfield relevant to this thesis. Throughout, we highlight how these fields contribute to designing intervention policies that can enhance the efficiency and effectiveness of business processes.

2.1. Machine Learning

Machine Learning (ML) is a subset of artificial intelligence that empowers systems to learn from data, enabling them to make predictions or decisions without explicit programming. The core objective of ML is to create systems that can automatically improve their performance through experience [7]. In the context of business processes, ML algorithms can be trained on historical instances of process executions to monitor ongoing cases and predict key measures of interest, such as delays or outcomes.

Here we review the key ML paradigms relevant to this thesis, including supervised learning, uncertainty estimation, reinforcement learning, and deep learning.

2.1.1. Supervised Machine Learning

Supervised machine learning is a fundamental approach where models are trained on labeled data pairs (X,Y) to create a mapping from input features X to known outputs Y. The main objective is to learn or approximate the relationship between the inputs X and the outputs y through a mapping function $f(X)$. Supervised learning tasks are generally divided into two main types: *classification* and *regression*.

In *classification* tasks, the output Y is categorical, representing distinct classes or categories [10]. For example, a classification model might predict whether an ongoing case will result in a positive outcome (e.g., customer satisfaction) or a negative outcome (e.g., customer complaint). In contrast, *regression* tasks involve predicting continuous numerical variables as the output Y [18]. An example could be predicting the remaining time until the completion of an ongoing case. This distinction between classification and regression is essential, as it influences the choice of algorithms and evaluation metrics used in the modeling process.

Here, $X = (x_1, \ldots, x_j)$ represents a j-dimensional vector of features (or independent variables) that describe ongoing cases at various points (or prefixes) in their execution, while Y denotes the target (or dependent) variable. The common workflow in ML tasks to learn the mapping function $f(X)$ can be divided into two main phases: *offline* and *online*.

During the *offline* phase, the data representing business process execution history is split into three subsets: *training*, *validation*, and *testing* sets. An ML algorithm is then trained and finetuned using the training and validation data. In the subsequent online phase, the trained and optimized model (predictive model) is evaluated on the testing set to assess its performance on unseen cases. Essentially, the predictive model is used to monitor ongoing cases that were not included in the training process at the operational level to predict $\hat{Y}$, the estimated value of Y.

Evaluation Measures. In the context of business processes, accurately estimating specific measures, such as outcomes or remaining time, is necessary, especially when these estimates inform proactive and timely business interventions. Consequently, evaluating the performance of a trained predictive model focuses on its accuracy in making predictions.

Various evaluation metrics are employed to assess predictive models on unseen operational data. The model predicts $\hat{Y}$ for each prefix of ongoing cases and compares it with the ground truth Y (the actual outcome observed in the data). A prediction that matches Y (or is close in regression tasks) indicates a reliable estimation. However, due to the inherent nature of predictive models, these estimations typically carry some uncertainty.

For classification tasks, metrics such as accuracy, precision, recall, and F1-score are utilized to evaluate the performance of classification tasks. *Accuracy* is effective when classes are balanced; however, in cases of class imbalance—where, for example, positive cases outnumber negative ones—accuracy can be misleading, selecting classifiers that predict the majority class (positive). *Precision* and *recall* provide more informative assessments in such scenarios: precision measures the accuracy of positive predictions, while recall quantifies the classifier's ability to identify all positive instances. These metrics can also be harmonized into the *F1-score*, which balances precision and recall to provide a single measure of model performance. A common technique for evaluating a classifier involves constructing a *Receiver Operating Characteristic (ROC)* curve and calculating the *Area Under the Curve (AUC)* to understand the model's ability to distinguish between positive and negative outcomes across various decision thresholds. A significant advantage of the AUC metric over accuracy and F1-score is its unbiased nature, even in highly imbalanced class distributions [24].

For regression tasks, metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2) are employed to evaluate the performance of predictive models [197]. *MAE* measures the average magnitude of prediction errors, providing a straightforward interpretation of prediction accuracy. *RMSE* gives greater weight to larger errors, making it sensitive to outliers.

R-squared quantifies the proportion of variance in the remaining time that can be explained by the input features, indicating the model's explanatory power.

Machine Learning Algorithms. In this thesis, we use ML algorithms to estimate both the remaining times and outcomes of cases. In the following, we give an overview of them.

- *Random Forest.* The Random Forest (RF) [19] algorithm is a versatile ML algorithm widely used for classification and regression tasks. It operates by constructing many decision trees during training and outputs the mode of the classes (for classification) or the mean prediction (for regression) of the individual trees. Each tree in the RF is trained on a subset of the data and features, a technique known as *bagging*, which helps to reduce overfitting and improve generalization. Tuning the Random Forest involves adjusting several hyperparameters, such as the number of trees (iterations), the number of features considered at each split, the maximum depth of the decision trees, and the size of the sampled subsets.
- *Gradient Boosted Trees.* Similar to RF, the Gradient Boosted Trees (GBT) [21] algorithm is used for classification and regression tasks and builds an ensemble of decision trees. However, it operates by sequentially constructing an ensemble of trees, where each tree attempts to correct the errors of the previous ones. This iterative process involves combining the predictions of multiple trees to produce a final prediction. The GBT algorithm uses a technique known as *boosting*, which focuses on improving model performance by giving more weight to misclassified instances in subsequent iterations.

  Examples of GBT algorithms include:

  - Light Gradient Boosting Machine (LightGBM): Designed for efficiency with large datasets and high-dimensional data [23].
  - Extreme Gradient Boosting (XGBoost): Known for its speed and performance, it is widely used in machine learning competitions and applications [20].
  - Categorical Boosting (CatBoost): Particularly effective with categorical features, offering robust handling of categorical variables [22]).

  Tuning GBT algorithms requires adjusting several hyperparameters, such as the number of trees (iterations), the learning rate, the maximum depth of the trees, and the minimum number of samples required to split an internal node.

In summary, supervised learning aims to generate predictions that closely align with Y (or are near to it in regression tasks), indicating reliable estimations. This thesis employs the aforementioned ML algorithms to develop predictive models for predicting outcomes and estimating remaining times. However, these predictions typically involve a degree of uncertainty, which will be discussed in the following section.

2.1.2. Uncertainty in Machine Learning

Uncertainty is critical in decision-making, especially when monitoring ongoing cases and making choices at each process stage. Incorrect decisions can have significant consequences in high-risk applications, such as business process monitoring, potentially leading to incorrect outcomes or financial losses [37]. By quantifying the uncertainty associated with a model's predictions, decision-makers can make more informed choices, thereby avoiding unnecessary or risky interventions.

For example, in a loan origination process, consider a predictive model that estimates the probability of a negative outcome (e.g., loan rejection) to be above 0.5 for a particular case at a specific point in time. This classification would lead to the preparation and sending of a second loan offer (an intervention) to the applicant. However, if the case were actually likely to accept the first loan offer (a positive outcome), this misprediction would result in wasted time, effort, and resources. This scenario illustrates the implications of low-quality predictions or those accompanied by high uncertainty. Consequently, managing uncertainty in prescriptive process monitoring involves understanding and quantifying uncertainties to ensure effective decision-making.

In the following, we will discuss two approaches to measuring uncertainty in model predictions: uncertainty quantification [37] and conformal prediction [41].

Uncertainty Quantification. Traditionally, uncertainty is modeled using probability theory, which has long been considered the primary method for addressing uncertainty in ML [38]. However, this probabilistic modeling approach, which relies on capturing learning through a single probability distribution, fails to distinguish between two fundamentally different sources of uncertainty: *aleatoric* [39] and *epistemic* [40] uncertainty. These two sources of uncertainty are combined to form the *total uncertainty* associated with predictions.

Aleatoric uncertainty, also known as data uncertainty, arises from the inherent variability in the data. It represents the natural randomness and noise present in the observations, which cannot be reduced by collecting more data. For example, the variability in customer behavior or fluctuations in market conditions contribute to aleatoric uncertainty [38]. In ML, this type of uncertainty can be modeled by considering the output distribution given the inputs, often using probabilistic models or Bayesian neural networks to estimate the confidence in individual predictions.

Epistemic uncertainty, or knowledge uncertainty, stems from a lack of knowledge about the best model to describe the data. This type of uncertainty can be reduced by gathering more data or improving the modeling techniques [38]. In business process monitoring, epistemic uncertainty might arise from limited historical data, inadequate feature engineering, or suboptimal model selection. Methods such as Bayesian approaches, ensemble techniques, and model averaging are employed to quantify and reduce epistemic uncertainty, thus enhancing the robustness of predictions.

Total Uncertainty (TU), composed of aleatoric and epistemic uncertainties, quantifies the uncertainty associated with predictions, typically using probabilistic models and ensemble techniques. Thus, not all ML algorithms can apply the total uncertainty estimation procedure. Additionally, total uncertainty in decision-making problems is often challenging to use and interpret, as it lacks a predefined confidence level guarantee. Hence, the *conformal prediction* framework is utilized to address these limitations.

Conformal Prediction. Conformal Prediction (CP) is a framework that transforms point estimates into prediction intervals for regression tasks or prediction sets for classification tasks, all based on a user-specified confidence level [41]. One of the key features of CP is that it is model-agnostic, meaning it can be applied independently of the underlying ML model. The CP framework employs nonconformity scoring measures along with calibration data to construct sets or intervals that encompass the true outcome with a defined significance level [130, 120, 121]. This characteristic provides finite-sample guarantees that the prediction set or interval will include the true outcome at the specified confidence level, in contrast to heuristic methods of uncertainty quantification [119, 126]. The user-defined significance level, denoted as α, directly influences this confidence level. A lower significance level correlates with a higher confidence level, which implies greater certainty about the model's predictions. For instance, if a user desires a 90% confidence level, they would set α to 0.1. However, higher confidence levels yield larger prediction intervals or sets to encompass all outcomes.

The operation of a conformal prediction technique involves two main steps. The first step entails calculating the nonconformity score for each case prefix in the calibration set using a nonconformity scoring method. This approach is referred to as *inductive conformal prediction* [128, 127, 129], which distinguishes it from *transductive conformal prediction* [130], the latter of which does not require a calibration set. For example, in regression tasks, the nonconformity score could be defined as the absolute difference between the predicted and actual values within the calibration set, thereby quantifying how unusual a given prediction is compared to the actual outcome.

The second step involves determining the nonconformity quantile, specifically the $(1-\alpha)$-quantile of the nonconformity scores in the calibration set. This quantile is established based on the predefined confidence level $(1-\alpha)$ and is utilized to define the prediction interval or set. The nonconformity quantile directly influences the dimensions or size of the prediction sets and intervals; a higher confidence level results in larger prediction sets or intervals to accommodate all possible estimates concerning outcomes or point estimates.

This thesis uses the conformal prediction framework on top of the ML models to calibrate them. This calibration enables us to make operational decisions based on estimates with specified confidence levels. By doing so, we enhance the accuracy and reliability of our decision-making processes, particularly in resource allocation and intervention strategies.

2.1.3. Deep Learning

Deep Learning (DL), a subset of machine learning, has made significant progress in recent years, particularly for tasks that involve large and complex datasets. DL methods such as transformers [322, 323], Long Short-Term Memory networks (LSTMs) [324, 325], and Graph Neural Networks (GNNs) [326, 327] have proven effective in modeling intricate temporal patterns, process structures, and heterogeneous event log data. While traditional ML methods remain central to this thesis, understanding DL's potential and limitations is important for providing context to the chosen methodologies and guiding future research directions. in PPM.

DL methods, especially in the field of process mining [332], excel in domains like time-series prediction [325, 328], next activity prediction [329, 330], anomaly detection [331], and outcome prediction [333] by learning directly from raw data without requiring extensive manual feature engineering. This ability allows DL models to outperform traditional ML techniques in tasks that deal with large-scale datasets containing complex relationships. However, these advantages come with trade-offs that need to be considered in practical applications.

One of the primary challenges of adopting DL is its requirement for vast amounts of labeled data and substantial computational resources [334]. Training DL models often requires large datasets and specialized hardware, such as GPUs or TPUs, which can be computationally expensive. This makes DL less suitable for environments with limited data or resources. In the context of this thesis, traditional ML methods are preferred because they have lower data and computational requirements.

2.1.4. Reinforcement Learning

Deep RL is a framework for developing decision-making agents that learn optimal behavior or policies by interacting with an environment through trial-and-error [42]. These agents receive rewards as feedback, aiming to maximize their cumulative reward over time. RL is grounded in the reward hypothesis, which asserts that all objectives can be framed as maximizing expected cumulative rewards.

RL algorithms are broadly classified into *model-based* and *model-free* approaches, differing in handling environmental dynamics and internal modeling [43]. *Model-based RL* algorithms employ explicit environment models to simulate potential future states and rewards. This method can enhance learning efficiency and decision-making accuracy, but it also necessitates accurate modeling of environmental dynamics. In contrast, *model-free RL* algorithms learn policies directly from interactions with the environment, relying solely on trial-and-error learning without explicit environmental modeling.

The application of RL in business process context mostly falls into the model-free RL algorithms. This is because model-free methods are more straightforward to implement and can be more robust in complex or stochastic environments, although they typically require more interactions to learn effectively. Further, model-free RL algorithms are classified into *value-based* and *policy-based* methods [158].

Value-based methods focus on estimating the value of different actions or states, aiming to find the optimal value function that maximizes cumulative reward. Examples include Q-learning [159] and deep Q-networks (DQN) [160]. On the other hand, *policy-based* methods directly learn the policy, which is a mapping from states to actions, without explicitly estimating the value function, e.g., Proximal Policy Optimization (PPO) [161], which is a simpler and more robust alternative to other policy optimization algorithms like Trust Region Policy Optimization (TRPO) [162]. PPO aims to improve the stability and reliability of policy updates in RL.

In the context of developing intervention policies during operational time, our focus is on *policy-based* RL algorithms such as *PPO*, which is considered an *online RL* algorithm, a policy optimization algorithm within the model-based category. In online RL, agents gather data directly from the environment by interacting with it, accumulating experiences in real time. These experiences are then used to update the agent's policy directly or through a replay buffer, enabling continuous learning and adaptation [44].

In practical applications like business processes, RL enables agents to make real-time decisions to guide ongoing cases toward a positive outcome. At each time step t, the agent observes the environment's state S_t, which represents an event occurrence in a specific case and encapsulates information about the case's current status at a given prefix. Based on this state, the agent selects an action A_t, forming the state-action pair (S_t, A_t). As the environment transitions to the next time step, $t+1$, the agent observes the new state $S_t + 1$ and receives a reward R_t based on the action taken from the state S_t. Through iterative learning and policy adjustment, the agent incrementally enhances its decision-making performance, adapting to varying environmental conditions and improving outcomes over time.

In the context of business processes, The application of RL in process mining has gained considerable attention in recent years, with a growing body of research demonstrating its potential to enhance various aspects of business processes [154]. Specifically, model-free RL algorithms (value-based and policy-based) have been effectively utilized for resource allocation and predictive and prescriptive monitoring of business processes.

RL has been applied to resource allocation in business processes to optimize the assignment of tasks to resources [152, 153]. By dynamically learning from the ongoing process data, RL-based systems can adjust resource allocation strategies to improve efficiency and productivity. For example, [150] introduced an RL approach for dynamic task assignment in workflow management, demonstrating significant improvements in resource utilization and process throughput. Additionally, the works in [151, 155, 156] addressed resource allocation optimization in business processes using a Q-learning algorithm, i.e., value-based RL algorithm, showing improved cycle time due to allocating resources efficiently.

Additionally, RL algorithms have recently been employed to enhance predictive monitoring by continuously learning from the process environment and improving prediction accuracy over time. For instance, [95] proposed an RL approach for predicting the next activity and remaining time in business processes via utilizing two RL agents simultaneously. Another example is provided in [94], where RL is used to predict the next activity to perform. This approach is extended in [157] to recommend activities related to decision-making along with identifying the most efficient path on a process model. These studies primarily use value-based RL methods, such as Q-Learning and Deep Q-Networks (DQN), which focus on learning the value function (e.g., Q-values) and derive the policy indirectly by selecting actions that maximize the value function. However, these methods can be unstable in large or complex state-action spaces.

2.2. Causal Inference

Causal Inference (CI) techniques go beyond identifying predictive relationships among variables to determine the independent effect of a phenomenon on an outcome of interest [25]. In the context of a loan application process, this means not just predicting whether a loan will be approved based on applicant characteristics but also understanding which factors causally influence the approval decision and estimating their effects on the outcome. Traditionally, CI has focused on inferring causation from Randomized Controlled Trials (RCTs). However, the field has increasingly focused on inference from observational data because RCTs are usually expensive or difficult to enforce.

There are two general tasks in causal inference: *discovery* and *estimation*. *Causal discovery* focuses on identifying the causal structure among variables, often through graphical models or algorithms that infer causation from manually constructed graphs, if they exist. This involves causal identification, where the main goal is to identify causal relationships between variables using a *causal graphical model framework* [26].

For instance, in [134], the authors present an approach based on time series analysis to automatically discover cause-effect relations between a range of process attributes and process performance indicators, such as cycle time. Additionally, the work in [135] introduces a systematic approach to discovering causal dependencies between events encoded in large data, called causality mining. In [136], the authors provide a framework to answer what-if questions about business processes by utilizing structural causal models to confirm existing cause-effect assumptions, control confounding, and answer intervention and counterfactual questions. Similarly, the work in [137] develops causal equation models for processes to identify attributes causing a problem and the effect of an intervention on any of the features. This work is further extended to include counterfactual explanations of case-level predictions [138].

Causal estimation, on the other hand, aims to quantify the effect of one variable (i.e., *intervention*) on another (i.e., *outcome*) automatically without relying on domain knowledge to construct causal graphs. For example, estimating the effect of an intervention, such as sending a second loan offer, on the loan application outcome. This typically involves regression analysis, propensity score matching, or instrumental variables to estimate causal effects from observational data, known as the *Neyman-Rubin potential outcomes framework* [27]. Since the main goal of this thesis is to develop intervention policies to provide prescriptive analytics based on event logs, we will focus only on causal estimation from observational data using the potential outcomes framework to estimate the causal effect automatically.

2.2.1. Potential Outcomes Framework

The first step in this framework is to identify an *intervention or a treatment* (T) that can positively impact the outcome (Y) of business processes. This intervention is an action or activity, such as "send a loan offer" or "reduce monthly interest," that, if performed, is expected to affect the outcome. It is captured by a binary variable $T \in \{0,1\}$, where $T = 1$ means the intervention is applied and $T = 0$ is not. Therefore, each case at different prefixes can be associated with two potential outcomes: the outcome that would have happened if the intervention was applied ($Y(T = 1)$) and the outcome that would have happened under no intervention ($Y(T = 0)$). However, it is worth noting that in any observational data, such as data representing business process execution history, only one of the two potential outcomes is observed, and the other is missing. For example, if a case receives a second loan offer, i.e., $T = 1$, we observe the outcome with the intervention, but the outcome without the intervention remains unobserved. Therefore, we can not observe both potential outcomes, which is the fundamental problem of causal inference [28].

The second step, given an observation data and an identified intervention (T), is to estimate the causal (or intervention) effect of the intervention (T) on the outcome (Y). Different estimates can represent the intervention effect, including Individual Treatment Effects (ITE) and Conditional Average Treatment Effects (CATE).

Definition 2.2.1. ***Individual Treatment Effect***: Let i indicate a business case. Then, the individual treatment effect (ITE) of a case i is defined as follows:

$$ITE_i = Y(T = 1)_i - Y(T = 0)_i \tag{2.1}$$

The ITE measures the discrepancy in outcomes for a specific case at a designated decision point when exposed to the intervention (T=1) versus when not (T=0), providing insights into the effect of the intervention on individual cases.

Definition 2.2.2. ***Conditional Average Treatment Effect***: Let i indicate a business case, and X_i be a set of attributes that characterize the case i. Then, the

Conditional Average Treatment Effect (CATE) of a case i is defined as follows:

$$CATE_i = \mathbb{E}[Y(T=1)_i - Y(T=0)_i \mid X_i] \tag{2.2}$$

Conversely, The $CATE$ focuses on the average or expected effect of the intervention on a subgroup of cases that share similar characteristics or conditions. It involves partitioning the dataset into groups based on certain attributes or conditions (X_i) and then calculating each group's average treatment effect, offering insights into how the intervention's impact varies across different subpopulations.

Generally, causal (or intervention) effect estimation methods in the potential outcomes framework can be categorized into two major groups [139]. The first group includes methods that extend supervised ML algorithms to estimate causal effects, mostly related to a family of methods known as *meta-learners*. These include single-model approach (S-learner) [140], two-model approach (T-learner) [141], X-learner [142], R-learner [143], Transformed Outcome Approach [144], and Deep Learning-based Methods [145]. The second group consists of methods specifically designed for estimating causal effects, which can be classified into four subgroups: Tree-based [146], Support Vector Machine (SVM)-based [147], Generative deep learning [145], and Ensemble-based methods [148, 149].

This thesis uses an ensemble-based method from the second group to estimate the causal effect. Most ensemble methods use tree-based methods, which build binary trees using specially designed splitting criteria to estimate the causal effect, as base learners. Typically, ensemble-based methods perform better than single tree-based models; however, ensemble-based methods lose interpretability and have higher time complexity than tree-based methods [139].

2.2.2. Realcause: Evaluating Causal Estimation Models

Observation data such as historical business process execution do not have ground truth for the intervention effect due to the fundamental problem of causal inference. This means that causal estimators cannot be evaluated like supervised ML models. Therefore, evaluating causal models using simulated or synthetic data where the true causal effects are known is common practice [29]. However, results derived from fully synthetic data might not generalize well to real-world data. To address this, semi-synthetic data can be used, where case attributes come from real data, but the treatment assignment and outcome are generated from arbitrary stochastic functions [30]. However, this procedure still needs more realistic treatment and outcome mechanisms.

A more realistic approach involves using a generative model to generate realistic synthetic data that is fit to real data, a method known as *Realcause* [24]. This model is fitted to real data, where case features, interventions, and outcomes are available. The fitted generative model is then used to sample data to simulate outcomes under both intervention and no-intervention conditions. This approach provides realistic data with the ground truth intervention effect while being statistically indistinguishable from the real data.

Realcause involves training models for intervention assignment given a set of attributes, i.e., $P(T|X)$, and outcomes given the intervention and case attributes, i.e., $P(Y|T,X)$. A neural network is used to model these mechanisms. The input of the neural network is X (the set of attributes), and the output is the probability of receiving the treatment. For the potential outcomes, the neural network structure follows the one proposed in TARNet [31].

In this setup, all data, regardless of treatment group, are input to the neural network to obtain a treatment-agnostic representation of the data. The network then splits into two parts: one for the treatment group, i.e., $T = 1$, and one for the control group, i.e., $T = 0$. Two more neural networks represent these two parts. Both take the treatment-agnostic representation as input, but one network uses only the data from the treatment group, while the other uses data from the control group. In Realcause, all three networks use the same architecture for simplicity.

The output of this setup is used to parameterize the distribution of the data, which is assumed to be the Bernoulli distribution. This distribution is sampled under both treatment and control settings to obtain realistic data with potential outcomes, i.e., $Y(T = 1)$ and $Y(T = 0)$. This approach allows us to create a realistic synthetic dataset where the true intervention effects are known, enabling the evaluation of causal estimation models and causal intervention policies.

2.3. Survival Analysis

Survival analysis is a set of statistical techniques used to analyze *time-to-event* data, focusing on measuring the duration from a defined starting point to a specific endpoint [32]. These techniques are typically applied to collected data over time, such as historical business process execution, where the time origin and event of interest must be clearly defined. Also, survival analysis is particularly relevant in business process management in scenarios where understanding the time-to-event is important for decision-making. For instance, estimating the time until a loan application is rejected in loan origination processes can help optimize resource allocation and prioritize intervention triggers. Here, the time origin could be when an applicant submits an application, and the endpoint might be the occurrence of a negative outcome due to canceling the loan offer.

Unlike the traditional remaining time prediction task, which focuses on estimating the time until the completion of an ongoing case regardless of its outcome, survival analysis specifically targets predicting the time until a negative event occurs, such as a loan rejection. This distinction is important as survival analysis deals with time-to-event data, where the event of interest (e.g., loan rejection) may not be observed for all cases. Hence, survival analysis deals with *censoring*, where some cases may not experience the event by the end of the process execution, introducing complexities in modeling.

Survival analysis often uses methods like *Kaplan-Meier estimation* [33],*Cox proportional hazards [34]*, or *parametric survival models* [35]. In contrast, remaining time PPM might use regression-based techniques or ML algorithms to predict continuous time durations. In this thesis, we use the Cox proportional hazards method, widely used in survival analysis for its flexibility—it does not require assuming a specific survival distribution [36]. Instead, these models utilize a *hazard function* to describe the rate at which events occur for cases that have not yet experienced the event. This approach allows us to investigate how case characteristics are associated with the time until an event occurs.

2.4. Process Mining

Process Mining (PM) is a set of data-driven techniques that analyze, monitor, and discover opportunities to enhance processes' performance [4]. A typical objective of performance improvement is to increase the rates of positive outcomes in business processes, such as boosting the number of applicants who accept loan offers in a loan origination process. Unlike traditional performance monitoring dashboards, which primarily focus on numerical aggregates of performance measures, process mining goes into the processes, activities, flows, and decisions to explain the underlying reasons for observed performance.

2.4.1. Event logs

At the core of any process mining technique is the *event log* [5] (see example in Fig. 3), which records the execution history of business processes within modern organizations. These processes are supported by process-aware information systems, which track and log their activities. For example, a loan origination process, also known as the application-to-approval process, a business case starts when a customer submits a loan application and continues until a decision is made. This process contains multiple events, activities, and decisions involving various participants and resources, eventually leading to an outcome of value for the organization or its customers, such as the customer accepting the loan offer and signing the contract with the bank. By leveraging these logs, organizations can optimize their business processes in a data-driven manner.

An event log consists of a sequence of traces or business cases. Each case comprises a series of events, and each event is characterized by a set of attributes (features) reflecting either the *control flow* and *data payload* of the process, as shown in Fig. 2. Both the control flow and data payload can include case (static) or event (dynamic) attributes. *Case attributes* are static throughout the case's lifetime, whereas *event attributes* change with each event.

Any event log must contain information about the control flow of the process, represented by three attributes: (1) *activity*, the performed action, such as creating an application or sending an offer; (2) *timestamp*, indicating when the activity occurred; and (3) *case ID*, a unique identifier for each case. These three attributes

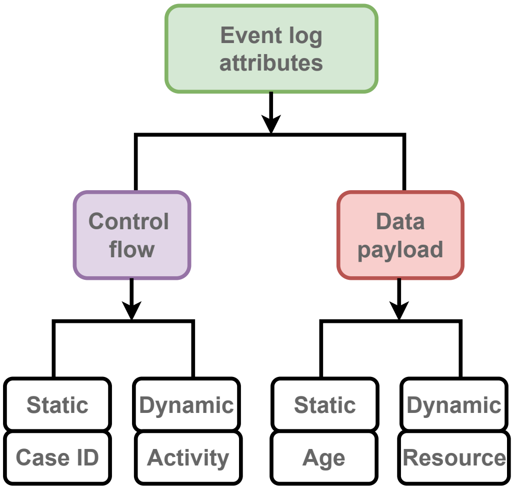
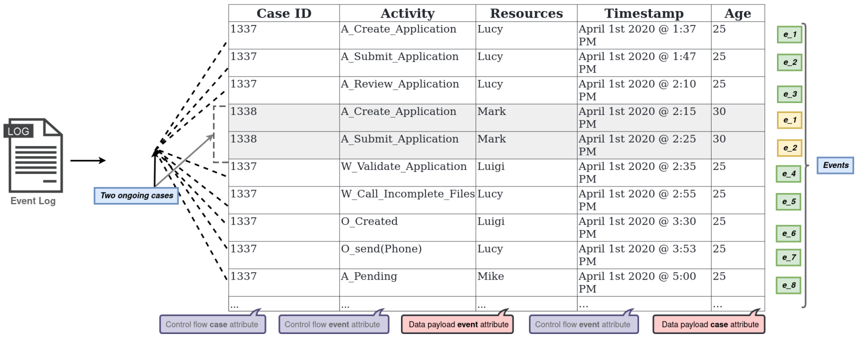


Figure 2. Event log attributes categorization.

collectively form an event. For example, in Fig. 3, event number one has a Case ID of 1337, the activity "create an application," and it occurred on April 1st at 1:37 pm. Here, the Case ID is static throughout the case's lifetime, making it a case attribute. In contrast, the timestamp and activity are dynamic attributes that change with each event, making them event attributes. Further, data payload attributes are optional in any event log. Case payload attributes, e.g., the applicant's age, remain constant, while event payload attributes, e.g., the resource (i.e., the process worker executing a specific activity), can vary throughout the case.



Case ID	Activity	Resources	Timestamp	Age
1337	A_Create_Application	Lucy	April 1st 2020 @ 1:37 PM	25
1337	A_Submit_Application	Lucy	April 1st 2020 @ 1:47 PM	25
1337	A_Review_Application	Lucy	April 1st 2020 @ 2:10 PM	25
1338	A_Create_Application	Mark	April 1st 2020 @ 2:15 PM	30
1338	A_Submit_Application	Mark	April 1st 2020 @ 2:25 PM	30
1337	W_Validate_Application	Luigi	April 1st 2020 @ 2:35 PM	25
1337	W_Call_Incomplete_Files	Lucy	April 1st 2020 @ 2:55 PM	25
1337	O_Created	Luigi	April 1st 2020 @ 3:30 PM	25
1337	O_send(Phone)	Lucy	April 1st 2020 @ 3:53 PM	25
1337	A_Pending	Mike	April 1st 2020 @ 5:00 PM	25
...	...	...	...	...

Figure 3. Example of an event log.

More formally, we assume that events are represented by diverse features (i.e., event and case attributes).

Definition 2.4.1. ***Event***: Assume E represents the universe of all possible events, and $\forall e \in E : e = a, c, t, (m_1, \ldots, m_n)$ where: a refers to an activity name; c is a unique identifier of a business case (or case ID); t refers to the timestamp; $m_1, \ldots, m_n$ represent all other attributes, where $n > 0$.

The sequence of events per a given case creates a trace. Formally,

Definition 2.4.2. ***Trace***: Let R be the universe of all traces. For $\forall \sigma \in R$, σ is a sequence of events such that $\sigma = (e_1, \ldots, e_n)$, where all events in σ are part of the same case.

A collection of completed traces, which record the execution of completed cases, constitutes an event log.

Definition 2.4.3. ***Event Log***: An event log L is a collection of complete business cases (or traces), such that $L = (\sigma_1, \ldots, \sigma_n)$.

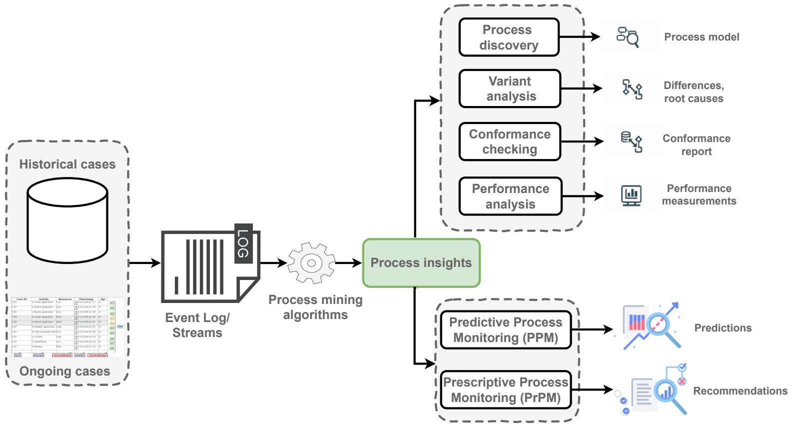


Figure 4. Overview of process mining techniques.

2.4.2. Process Mining Capabilities

Modern organizations use process mining techniques with an event log as an input to solve various tasks [4]. Examples of process mining techniques include (see Fig. 4):

- *Automated process discovery* involves creating a process model that defines how a given process was actually executed, known as an as-is process model. This model may be used later to compare it to the intended process structure.
- *Conformance checking* compares the extracted event log with a process model to identify where cases deviate from the expected or ideal process.
- *Performance analysis* determines the reasons for observing a particular performance and how to improve it. For example, it identifies bottlenecks in the two-week loan processing time and suggests ways to overcome them.
- *Variant analysis* examines how a specific process is carried out under different conditions. For instance, one variant may consist of cases with positive outcomes, while another may involve cases with negative outcomes. This type of analysis assists in identifying the reasons for suboptimal performance in the process.

The abovementioned techniques do not help in monitoring ongoing cases in run-time (or operational time) to take proactive action, if needed, and avoid negative outcomes. To address this, two related process mining techniques have been introduced:

- *Predictive process monitoring* aims to predict the unfolding of ongoing cases, including determining whether they will result in a positive or negative outcome. However, simply predicting negative future outcomes is not as valuable as being able to influence those negative outcomes positively. This is where prescriptive process monitoring comes into play.
- *Prescriptive process monitoring* goes beyond predictions by recommending actions or interventions to improve specific measures of interest. For example, it analyzes the potential impact of triggering an action, such as determining if sending another loan offer to an applicant will increase the probability of the offer being accepted.

This thesis aims to implement prescriptive process monitoring techniques to develop intervention-based recommendations to optimize business process performance by increasing the number of cases that result in positive outcomes.

2.5. Predictive Process Monitoring

PPM involves forecasting the future state of ongoing cases in a business process using incomplete execution traces and historical logs of completed traces. It encompasses a range of techniques designed to monitor and predict the outcomes or progression of ongoing process instances. For example, predictive monitoring can be used to predict case outcomes (e.g., positive or negative) [10], the remaining time until a case is completed [18], or the next activity to be performed within the process [198].

In this thesis, the focus is on predicting case outcomes and remaining time to support the development of intervention policies, where interventions are triggered in a timely manner to prevent costly negative outcomes.

Within the context of business processes and during operational time, we monitor incomplete (ongoing) cases. The objective at any given time is to estimate specific measures of interest based on the available prefix of a case, such as the outcome. To simulate incomplete cases from the completed ones, which may come with different lengths, we use a *prefix function* to extract all possible prefixes up to a certain length, k, and the output from this step is a *prefix log*.

Definition 2.5.1. ***Prefix function*** P: Given a complete trace $\sigma = (e_1, .., e_n)$ and a prefix length k, where $0 \leq k \leq n$, then $P(\sigma, k) = (e_1, ..., e_k)$

Given the prefix log, we train ML algorithms on case prefixes to simulate real-life situations. In practice, during run-time, we receive the first event of a case and estimate a measure of interest with a prefix length of 1. With each subsequent event, the prefix length increases (e.g., 2, 3, ..., k), and the task is to estimate the measure of interest accordingly. Therefore, the input for any ML technique is a case prefix up to the length of k, which typically includes attributes (features) derived from event logs, either originally present or engineered, and can have various data types, such as categorical or numeric. Hence, we use a *sequence encoding* technique to transform prefix attributes into a fixed-size feature vector, where all case prefixes have the same length, suitable for training ML techniques.

2.5.1. Feature encoding for Predictive Process Monitoring

Feature encoding in PPM is a step that involves transforming the data from process event logs into a format that ML models can effectively utilize. Since the execution traces of business processes vary in length, encoding methods are required to convert these traces into fixed-length vectors, allowing the ML algorithms to process and learn from them.

Definition 2.5.2. ***Sequence Encoder** enc*: Given a prefix $P(\sigma,k) = (e_1,...,e_k)$, the sequence encoder *enc* maps the prefix to a fixed-size feature vector $enc(P(\sigma,k)) = \mathbf{x}$, where $\mathbf{x} = (x_1,...,x_j)$ is a j-dimensional feature vector containing encoded features from the prefix.

Table 1. Sample case extracted from the event log.

Case Attributes		Event Attributes		
Case ID	Age	Activity	Timestamp	Resources
1337	25	A_Create_Application (1)	April 1st 2020 @ 1:37 PM	Lucy
1337	25	A_Submit_Application (2)	April 1st 2020 @ 1:47 PM	Lucy
1337	25	A_Review_Application (3)	April 1st 2020 @ 2:10 PM	Lucy
1337	25	W_Validate_Application (4)	April 1st 2020 @ 2:35 PM	Luigi
1337	25	W_Call_Incomplete_Files (5)	April 1st 2020 @ 2:55 PM	Lucy
1337	25	O_Created (6)	April 1st 2020 @ 3:30 PM	Luigi
1337	25	O_send(Phone) (7)	April 1st 2020 @ 3:53 PM	Lucy
1337	25	A_Pending (8)	April 1st 2020 @ 5:00 PM	Mike

Many sequence encoding techniques have been introduced in the literature [10] to handle static case attributes and dynamic event attributes, which can vary in data types, such as numeric or categorical. These techniques include static encoding for case attributes and techniques like *last-state*, *aggregate*, *index*, and *combined* encoding for event attributes. Initially, the encoding process involves applying static encoding to case attributes. Then, a selected technique—such as last-state, aggregation, index, or combined encoding—is chosen to encode event attributes. These encoded representations are then combined to form the encoded prefixes. In Table 1, we extracted business case ID 1337 from the event log shown in Fig. 3 to illustrate these encoding techniques, using a prefix length $k = 3$ for simplicity.

- *Static encoding* focuses only on encoding case attributes "as is." It excludes the Case ID, which acts as a unique identifier and does not contribute additional information for modeling purposes. It employs one-hot encoding for categorical data, while numeric data remains unchanged due to its constant values throughout the case [9].

- *Last-state encoding* focuses on encoding event attributes using data from the most recent event occurrence [14]. This results in a feature vector size dependent on the number of event attributes. It discards all information preceding the last event, condensing only the latest event or state information. However, this encoding technique can be extended to include the last m states instead of just the last state, where $m = 1$. Table 2 illustrates this technique with a prefix length $k = 3$.

Table 2. Feature vectors with last-state encoding ($k = 3$) from Table 1.

Case Attributes	Event Attributes		
Age	Activity_last	Timestamp_last	Resources_last
25	A_Create_Application (1)	0	Lucy
25	A_Submit_Application (2)	10	Lucy
25	A_Review_Application (3)	23	Lucy

- *Aggregate encoding* differs from last-state encoding by considering information from all events since the beginning, thus avoiding information loss inherent in the last-state encoding technique. Aggregate encoding employs various aggregation functions to have a fixed-size feature vector. These functions typically include minimum, average, maximum, and sum of observed values for numeric data. The count function is commonly used for categorical data, such as counting the occurrences of activities [15]. Table 3 illustrates this technique with a prefix length $k = 3$.

Table 3. Feature vectors with aggregate encoding ($k = 3$) from Table 1.

Case Attributes	Event Attributes				
Age	Activity_1	Activity_2	Activity_3	Timestamp_sum	Resources_Lucy
25	1	0	0	0	1
25	1	1	0	10	2
25	1	1	1	33	3

- *Index encoding* complements aggregate encoding by preserving the order of event occurrences, thus avoiding any information loss. For each executed event (index), features are generated for each event attribute, including its order. This encoding technique is lossless, capturing all possible information that can enhance the training process of ML algorithms. However, it results in high-dimensional feature vectors due to the detailed representation of each event attribute, especially with longer prefixes [16]. Table 4 illustrates this technique with a prefix length $k = 3$.

Table 4. Feature vectors with index encoding ($k = 3$) from Table 1.

Case Attributes	Event Attributes								
Age	Activity_1	Timestamp_1	Resources_1	Activity_2	Timestamp_2	Resources_2	Activity_3	Timestamp_3	Resources_3
25	A_Create_Application (1)	0	Lucy	A_Submit_Application (2)	10	Lucy	A_Review_Application (3)	23	Lucy

- *Combined encoding* gives the ability to integrate multiple event attribute encoding techniques. For example, it integrates the last-state and aggregate encoding techniques to provide a balanced view of both recent and cumulative process behaviors.

The sequence encoding techniques mentioned above are employed to extract features from a trace that encode information regarding the actions performed during the execution of a trace, the order of these actions, and features related to event or case attributes. These features capture both the control flow and contextual information of the process execution.

In recent years, DL-based encoding methods have gained attention for their ability to learn complex patterns in structured data. Techniques such as sequence encoding (e.g., using LSTMs) [325], graph encoding (e.g., using GNNs) [189], and imagery encoding (e.g., transforming event logs into image-like structures for CNNs) [335] have shown promising results in tasks like next-activity prediction. However, this thesis focuses on outcome-oriented benchmarking using traditional ML methods [228], prioritizing simplicity and resource efficiency over the complexity of DL techniques.

Once the trace is encoded into a fixed-length feature vector, this vector is used as input for training ML models for two main PPM tasks: *outcome-oriented* and *remaining time*, which will be discussed in the following subsections.

2.5.2. Outcome-Oriented Predictive Process Monitoring

An outcome-oriented PPM task focuses on predicting whether a case will end with a positive or negative outcome, defined by specific business objectives [10]. In a loan origination process, for instance, a positive outcome is achieved when the loan application is accepted, the applicant agrees to the terms and signs the contract with the bank. Conversely, a negative outcome may arise if the applicant declines the offer or the bank cancels the application due to incomplete documents. This binary classification task is central to our thesis, focusing on developing a probabilistic classification model. This model identifies cases at risk of a negative outcome by predicting probability scores above a decision threshold. For example, a probability score above 0.5 for a negative outcome defines it as a negative case. Such models are evaluated by comparing model predictions with actual outcomes from event logs to ensure high accuracy and reliability in decision-making contexts.

Definition 2.5.3 (Outcome-Oriented Probabilistic Classifier)**.** An outcome-oriented probabilistic classifier $\text{cls}_{\text{outcome}} : X_1 \times \cdots \times X_p \to [0,1]^2$ is a function that takes a p-dimensional encoded sequence of events $(X_1, \ldots, X_p)$, corresponding to the sequence of activities in a business process case, and outputs a probability distribution over the class labels $Y \in \{0,1\}$. The output of the classifier is a pair of probabilities $(\hat{p}_0, \hat{p}_1) = \text{cls}_{\text{outcome}}(X_1, \ldots, X_p)$, where:

- $\hat{p}_0$ is the probability that the case will have a negative outcome $(Y = 0)$,
- $\hat{p}_1$ is the probability that the case will have a positive outcome $(Y = 1)$, such that $\hat{p}_0 + \hat{p}_1 = 1$.

2.5.3. Remaining Time Predictive Process Monitoring

Unlike outcome-oriented PPM, remaining time PPM estimates the time remaining until the completion of an ongoing case [18], regardless of its eventual outcome. This task helps prioritize cases that need intervention, allowing for timely resource allocation and effective management. For instance, if two cases are predicted to have negative outcomes in a loan origination process, but one will end within 5 days and the other within 10 days, it is more appropriate to intervene in the first case sooner. The predicted remaining time is evaluated against the actual time observed in the event logs to ensure the model's reliability and accuracy.

Definition 2.5.4 (Remaining Time Regressor)**.** A remaining time regressor $\text{reg}_{\text{time}} : X_1 \times \cdots \times X_p \to \mathbb{R}_{\geq 0}$ is a function that takes a p-dimensional encoded sequence of events $(X_1, \ldots, X_p)$, representing the sequence of activities in an ongoing case, and estimates the remaining time $r \in \mathbb{R}_{\geq 0}$ until the case reaches its conclusion. The remaining time regressor outputs a prediction $\hat{r} = \text{reg}_{\text{time}}(X_1, \ldots, X_p)$, which corresponds to the predicted time left before the case ends.

It is worth noting that traditional PPM techniques primarily focus on maximizing prediction accuracy, often without addressing the quality or reliability of these predictions. For example, it can be useful to predict whether ongoing cases will end with a negative outcome or estimate their completion within a specific timeframe (e.g., five days). However, the reliability of such predictions depends on the quality of the underlying data and models. Low-quality predictions can reduce certainty and lead to poor decision-making. In the context of intervention policies, triggering actions based on uncertain or unreliable predictions, such as a probability score of a likely negative outcome, can result in inefficient resource allocation and suboptimal outcomes. Therefore, the following section discusses the role of uncertainty in PPM.

2.6. Uncertainty in Predictive Process Monitoring

In this section, we explore the role of uncertainty in the field of PPM, as it is directly relevant to the focus of this thesis. Recent PPM techniques focus on obtaining highly accurate predictions, often ignoring the uncertainty or quality of these predictions [99, 100, 101, 102, 103, 108, 109, 110, 112, 10, 18]. Accounting for uncertainty in predictions is necessary, especially if it is followed with action. For example, consider a scenario where a case is predicted to end with a negative outcome with a high probability at an early stage of its execution, leading a process worker to trigger and execute an intervention. However, if the case was actually going to end with a positive outcome and the predictive model made an incorrect prediction, the result would be wasted time, effort, and resources due to high prediction uncertainty. Therefore, focusing solely on developing highly accurate PPM techniques without quantifying uncertainty is impractical and can lead to inefficient resource utilization.

As discussed in Section 2.1.2 there are two main approaches to account for uncertainty in predictive models: uncertainty quantification and conformal prediction. We differentiate between these approaches in the following subsections.

2.6.1. Uncertainty Quantification

Uncertainty quantification in PPM can be achieved through Bayesian [122, 123] or ensemble [124, 125] methods. Ensemble methods are considered more computationally efficient. They often require less time and resources due to their ability to leverage parallelism and simpler aggregation techniques compared to Bayesian methods, which involve more complex probabilistic computations [123, 113, 114, 115]. Additionally, ensemble methods offer more reliability for out-of-distribution data, effectively handling inputs significantly different from the trained data distribution [113]. However, these uncertainty quantification methods deliver a heuristic quantification of uncertainty. Thus, they do not give any confidence level guarantees about model predictions [133, 119, 126] and thus may not be practical for decision-making.

PPM encompasses three general prediction tasks: outcome-oriented, remaining time, and next activity prediction. Uncertainty quantification in the remaining time prediction task is explored in [116], where a deep learning approach using concrete dropout, a variant of Monte Carlo dropout, is introduced to estimate uncertainty within neural networks. Two works have addressed uncertainty quantification for the next activity prediction task [117, 118]. The first employed the Monte Carlo dropout method, while the second discussed uncertainty quantification using Bayesian networks. This Bayesian approach directly learns the process topology from event logs, ensuring a transparent and explainable predictive process model. Nevertheless, quantifying uncertainty in the task of outcome-oriented PPM remains an area yet to be explored.

2.6.2. Conformal Prediction

Recently, conformal prediction techniques have gained attention in process mining, particularly in the PPM field. This enhances the reliability and accuracy of predictions by generating prediction sets or intervals that maintain a specified coverage rate. For instance, in [131], the authors demonstrate how conformal prediction can be integrated into a remaining time prediction task in PPM. By employing this approach, they produce not only point predictions but also intervals that encapsulate the true values, ensuring that a defined proportion of actual values falls within these intervals. This capability is particularly valuable in PPM, providing greater confidence in predictions and facilitating more informed decision-making.

The use of conformal prediction to other PPM tasks—such as outcome-oriented and next-activity prediction—remains largely unexplored. This gap is especially critical in high-risk applications, where PPM predictions directly inform subsequent actions. Enhancing the certainty of these actions is essential for effective resource allocation and process management. Furthermore, by integrating conformal prediction into PPM frameworks, organizations can improve their ability to navigate uncertainties in process execution, ultimately leading to enhanced overall performance and more favorable outcomes.

While PPM tasks offer valuable insights, their practical utility is limited if organizations are unable to influence the predicted outcomes. For instance, knowing that an applicant is likely to reject an offer within five days is helpful, but the predictive insight remains of limited value without the ability to intervene and change this outcome. The emerging field of prescriptive process monitoring extends beyond mere prediction to address this limitation. It aims to provide actionable recommendations and trigger interventions that can positively influence the course of events and enhance process outcomes. Thus, the focus shifts from simply predicting future events to actively influencing and improving them.

2.7. Summary

This chapter explored the interdisciplinary background of prescriptive process monitoring, which integrates concepts from machine learning, static analysis, and process mining to design effective intervention policies.

We discussed key machine learning concepts, focusing on supervised and reinforcement learning techniques. Given their relevance to this thesis, two statistical frameworks—causal inference and survival analysis—were also presented. The significance of these methods was highlighted, particularly in their application to business process tasks. Specifically, supervised machine learning is employed to predict case outcomes (positive or negative) and estimate the remaining time until completion. Additionally, survival analysis is used to estimate the time until the occurrence of negative events, causal inference is used to assess the impact of interventions, and online reinforcement learning is used to optimize intervention policies adaptively.

We then introduced core process mining concepts, such as events, traces, and event logs, which are foundational to the following analysis. Various process mining techniques applicable to tactical and operational settings were reviewed, particularly emphasizing the role of prescriptive process monitoring in these contexts. Additionally, we explored two key predictive process monitoring tasks: outcome prediction and remaining time estimation. In this exploration, we highlighted the importance of uncertainty in these tasks and its significance in developing effective intervention policies.

This thesis focuses on developing intervention policies that optimize business process performance, i.e., increasing the number of cases ending with a positive outcome. Integrating diverse learning techniques, particularly in environments characterized by resource constraints and uncertainty, ensures timely and effective interventions, thereby maximizing positive outcomes.

3. STATE OF THE ART

As discussed in Chapter 2, simply predicting how ongoing cases will unfold has limited utility without the capability to influence them positively. This is where prescriptive process monitoring becomes necessary. Unlike predictive process monitoring, which primarily forecasts future states, prescriptive process monitoring goes further by recommending or prescribing targeted interventions intended to influence ongoing cases positively [45]. Thus, the focus shifts from passively foreseeing future states to driving them toward more desirable results.

Based on the literature, this chapter categorizes the field of PrPM according to three main criteria: *optimization objective*, *intervention type*, and *policy learning type*. This categorization helps determine the thesis's scope and drives the discussion of the state-of-the-art around the key gaps identified in Section 1.1.

3.1. Prescriptive Process Monitoring

Prescriptive process monitoring (PrPM) techniques aim to develop an *intervention policy* (IP) that determines whether and when an intervention should be triggered in a process to optimize specific objectives [45]. Depending on the objective, the recommended or prescribed intervention varies to meet the desired outcomes. For instance, if the goal is to optimize resource allocation, the intervention type should be relevant to resource management. Additionally, how the policy is learned and updated during operational time is important; it can be based on predefined decision rules or learned automatically. Therefore, we categorize PrPM techniques according to *optimization objective*, *intervention type*, and *policy learning*.

3.1.1. Optimization Objective

PrPM techniques aim to optimize two main objectives: *process outcomes* and *performance metrics*.

- *Process Outcomes:* This objective focuses on increasing the number of cases that end with a positive outcome. Process outcomes are typically binary, with 1 indicating a positive outcome and 0 indicating a negative outcome. The targeted outcome type can vary: (1) *Categorical Outcomes:* The objective here is to avoid or mitigate undesired outcomes, such as reducing the occurrence of negative outcomes [61, 341, 46, 59, 47, 48, 49, 50, 51, 52, 53, 54]. (2) *Temporal Outcomes:* This focuses, for instance, on meeting deadlines and avoiding violations, ensuring that processes are completed within the desired time frames [55, 56, 57, 58, 59, 60].
- *Performance Metrics:* This objective is concerned with optimizing quantitative metrics of process performance. Performance metrics can be either minimized or maximized, depending on the specific metric. For example, minimizing cycle time or maximizing compliance. Most studies focus on

temporal objectives, such as reducing remaining or cycle time [70, 63, 62, 64, 65]. Nevertheless, other metrics have also been studied, including processing time [66], customer lifetime value[67], revenue growth [67, 68], and perceived service quality [69].

This thesis focuses on improving *process outcomes* where the outcome is categorical and binary, i.e., positive or negative. While it builds on work such as [61, 59, 46, 54], it differs by emphasizing the development of intervention policies that not only aim to improve process outcomes but also account for uncertainty and operate under resource constraints.

3.1.2. Intervention Type

An *intervention* is an action represented by activities identified in the event log from historical process executions and designated by domain experts as a potential means to improve outcomes. For example, in a loan origination process, an intervention might involve reducing the monthly interest rate to increase the likelihood of loan offer acceptance, thereby transforming a negative outcome into a positive one. Such interventions should effectively change the outcome status; without them, the outcome would likely remain negative. Additionally, interventions are designed to be executed under specific circumstances within the process, such as when a case is at risk of concluding negatively. These actions drive the process toward desired results by influencing critical decision points.

In the literature on PrPM, interventions are categorized into two main types: discrete and continuous [45].

- *Discrete Interventions:* These interventions are triggered at specific decision points, for example, when a negative outcome is expected. Here, PrPM methods prescribe actions that may or may not be executed to convert a negative outcome into a positive one. For instance, alarms can be triggered to indicate when a particular case requires attention, specifying who needs to intervene. However, the exact nature of the intervention is not predetermined, allowing the process worker to assess the situation and decide on appropriate actions based on established thresholds [47, 46, 48]. Another example could involve triggering a specific action, e.g., sending an alternative loan offer to an applicant to improve the acceptance rate [85, 61]. Nevertheless, limitations regarding how often alarms or prescribed actions can be triggered, the availability of resources for immediate action, and the criteria for defining cases in need of intervention can affect the overall effectiveness of these approaches.
- *Continuous Interventions:* In contrast, continuous interventions provide guidance rather than specific actions. They are activated at each decision point, recommending the next best activity to achieve specific performance metrics [60, 71, 72, 73, 69] or suggesting which resources are responsible for executing particular tasks [63, 56]. These interventions aim to optimize

outcomes by modifying the sequence or flow of process activities [74, 75]. They may suggest prioritizing critical activities, reducing cycle times, or ensuring compliance with established rules and regulations. Additionally, continuous interventions can involve reallocating resources to critical tasks or high-priority cases to speed up processing or adjusting resource availability based on demand changes to ensure timely task execution.

This thesis focuses on *discrete interventions* designed for resource-limited scenarios, considering a single intervention type that can be applied only once per case. Therefore, the decision is whether to execute the intervention, assuming that no further action will be possible for that case once it is applied. While similar studies in the literature also use a single type of intervention, they generally assume unlimited resources, enabling multiple interventions per case [341, 59, 46]. In contrast, this thesis addresses resource constraints, ensuring that intervention policies are aware of practical resource limitations.

3.1.3. Policy Learning Type

The core of a PrPM approach is an *intervention policy*: a decision-making procedure that identifies, during process execution, which process instances require intervention, determines the optimal timing for it, and allocates the appropriate resource (e.g., a process worker) to carry it out. Thus, the intervention policy is designed to address three key questions: Who requires intervention? When the intervention should be triggered? And who should execute it? Hence, at runtime, it ensures interventions are timely and efficient while considering resource constraints.

There are two main approaches to learning intervention policies in the literature: *White-Box* [46, 47, 70] and *Black-Box* [341, 59, 164, 164]. Each approach applies different decision-making strategies for triggering interventions. Next, we will explore how these learning approaches are used to create intervention policies.

White-Box Approach. This approach uses a rule-based framework, allowing process analysts to create transparent and configurable intervention policies. The "white-box" characteristic refers to the transparency in developing intervention policies during process execution rather than the underlying techniques used. It relies on predefined decision rules derived from domain expertise, historical data analysis, or both. These rules are interpretable, making it easier for process workers to understand and explain intervention policies. However, rule-based approaches can be inflexible and often difficult to adapt to dynamic process changes or unexpected scenarios.

Decision rules use specific estimates, reflecting the current state of cases within a process, to determine whether and when an intervention should be triggered under certain conditions [45]. These rules can vary based on *predictive*, *similarity*, *explainability*, or *causal* estimates. For instance, a predictive-based decision rule might involve using an estimate like the probability of a negative outcome. The condition for triggering an intervention could be triggering an immediate intervention when this probability exceeds a predefined threshold [46, 47].

Similarity-based rules compare the current case with past cases having similar characteristics or historical patterns [52, 71, 77, 50, 51, 78, 79, 80, 68, 75, 74, 81, 76]. If a current case resembles past cases that ended negatively under similar conditions, an intervention could be triggered to mitigate risks based on these similarities.

Explainability-based policies employ methods to increase trust by explaining why an intervention is recommended given a particular case, enhancing understanding and acceptance of the decision [87, 88].

Causality-based rules focus on identifying cause-and-effect relationships within the process. For example, an intervention might be triggered if certain actions within the process are likely to lead to a positive outcome [61, 82, 83, 84, 85, 86].

While decision rules based on these estimates ensure interventions are triggered based on quantifiable estimates, their quality varies, leading to *uncertainty* in decision-making. Moreover, rule-based approaches often assume unlimited resources for immediate intervention across all cases, which is impractical. Therefore, optimizing resource allocation becomes important to maximize the impact of interventions on process outcomes, prioritizing cases based on available resources and the urgency of intervention needs.

Black-Box Approach. This approach is considered as learning-based or data-driven, which primarily uses reinforcement learning algorithms [95] to develop intervention policies during operational time, i.e., *online-RL* [89]. These approaches can learn from data and adapt to new process states, making them more flexible and potentially more effective in dynamic environments. Learning-based approaches can uncover complex patterns and relationships in the data that rule-based approaches might miss. However, they often operate as black boxes, providing less transparency in decision-making, which can be a drawback in contexts where interpretability is important.

In contrast to rule-based approaches, learning-based approaches do not rely on predefined decision rules or thresholds for estimates to determine whether and when to trigger interventions. Instead, an RL agent is provided with information, such as estimates reflecting the current state of cases, and is required to act accordingly. These estimates could be *predictive-based*, *uncertainty-based*, or *causality-based*, like rule-based approaches. However, unlike rule-based approaches, the RL agent discovers the optimal timing and cases for triggering interventions automatically based on a feedback reward method. The RL agent can choose to intervene immediately, delay intervention to a later stage in a case's progression, or decide that intervention is unnecessary. This adaptive learning process allows the RL agent to optimize intervention policies dynamically, potentially leading to more effective decision-making in real-time and dynamic environments.

Among the different types of RL policies—predictive-based, uncertainty-based, and causality-based—each provides distinct information to guide the RL agent in learning effective intervention strategies. *Predictive-based* policies focus on outcome prediction, feeding the agent with data on the probability of a negative outcome for ongoing cases [92, 59, 90]. *Uncertainty-based* policies supplement this with information on the reliability of these predictions, ensuring interventions are triggered only when predictions are sufficiently certain [59, 93]. *Causality-based* policies go further by incorporating insights into the effectiveness of potential interventions, enhancing the agent's ability to make decisions that are not only timely but also likely to produce desired outcomes [86, 341].

In all the RL policies discussed, the agent relies on estimates reflecting each case's current state. During operation, historical cases are streamed event by event to the RL agent, providing information about the case history, predictions, uncertainty levels, or causal relationships. However, this approach presents two primary limitations. Firstly, the RL algorithm updates its policy on a case-by-case basis, potentially leading to data leakage issues common in business processes where cases often overlap. Secondly, these policies assume unlimited resources, allowing the RL agent to trigger interventions for multiple cases as needed. In reality, resources are finite, restricting the agent's capacity to trigger interventions for every case, highlighting a practical challenge in implementation.

Based on the above-mentioned categorization of PrPM by optimization objectives, intervention types, and policy learning, we narrow the problem addressed in this thesis. Specifically, we focus on improving process outcomes by assuming that each case resulting in a negative outcome incurs a cost. We aim to minimize this cost by triggering a single type of discrete intervention at run-time. During process execution, a process worker may choose whether to trigger the intervention; if implemented, this action reduces the cost associated with the negative outcome. We explore this problem within the setting of limited resources, where triggering the intervention also incurs its own associated cost. We examine this in both white-box and black-box settings. In the following section, we discuss work related to this problem.

3.2. Uncertainty in Prescriptive Process Monitoring

This thesis addresses the problem of improving process outcomes in scenarios where the outcome is binary, classified as either positive or negative. Although several studies have investigated methods for enhancing process outcomes [61, 59, 46, 54], they often ignore the importance of uncertainty awareness and the constraints imposed by limited resources. These approaches assume unlimited resources, implying that interventions can be executed without limitations. Consequently, they do not account for the reality of resource constraints, which means that only a subset of cases can receive interventions at any given time.

Awareness of uncertainty is necessary for informed decision-making, yet only two existing studies within the PrPM field explicitly utilize a measure of uncertainty to guide intervention decisions. For instance, the work in [59] assesses the reliability of outcome predictions through an ensemble of deep learning classifiers across various process states. Rather than directly quantifying uncertainty, this method evaluates prediction reliability based on the degree of consensus among the classifiers. A high level of agreement produces a high-reliability estimate, while divergence leads to a lower estimate. In conjunction with the predicted probability of a negative outcome, this reliability metric informs the intervention policy and determines whether to intervene in a black-box setting.

Additionally, the research presented in [61] utilizes conformal prediction to develop an intervention policy by providing a reinforcement learning (RL) agent with a prediction set instead of merely the probability of a negative outcome. This approach allows the RL agent to implement interventions with greater certainty. However, this study also operates under the assumption of unbounded resources, which is impractical in real-world scenarios. Furthermore, the work discussed in [132] explores the application of conformal prediction for automated decision-making in goodwill checks within the automotive industry, utilizing a goodwill dataset from a car manufacturer.

In this thesis, we consider the approaches outlined in [61, 59, 46] as baselines, as they address aspects of the problem at hand. However, we adapt these approaches to operate within limited resources.

3.3. Causality in Prescriptive Process Monitoring

The value of causality in PrPM arises from its ability to estimate the potential effects of interventions on process outcomes. This estimated effect determines whether to trigger an intervention in a white-box [70] or black-box [341] context.

PrPM proactively improves future outcomes by leveraging predictive and causal models to prescribe actionable interventions that optimize processes. For example, in loan origination, a PrPM system might recommend adjusting loan terms before a rejection occurs—such as offering a smaller loan amount or flexible repayment options—to increase approval likelihood. This forward-looking approach prioritizes improving future decisions through targeted interventions. The emphasis is on guiding future actions and decisions, focusing on what to do and when to achieve desired outcomes.

Before 2020, the literature on PrPM, as noted in [45], did not explore causal inference for optimizing process outcomes. However, a recent study in [85] introduced causal inference into process mining by employing association rule mining to identify potential interventions, addressing another problem, i.e., discovering potential interventions. They utilized an uplift tree to estimate causal effects. Additionally, [70] proposed a PrPM method that leverages causal inference to optimize business processes' cycle time, focusing on optimizing performance metrics rather than process outcomes. Further contributions in [61, 341, 86, 199] also discuss applying causal inference to optimize process outcomes. Notably, [86] compared causal inference, including counterfactual explanations, with RL. The work in [199] discussed the impact of learning causal relationships between activities in process monitoring settings.

While causal inference aims to optimize process outcomes by estimating the effects of potential interventions, counterfactual explanations take a different approach by retrospectively analyzing past outcomes. Instead of guiding future decisions, they seek to understand how alternative scenarios could have influenced a given decision [136, 320, 321]. Counterfactuals analyze "what if" situations to explain how different decisions could have led to different results. For instance, a counterfactual might reveal, “Had the applicant’s credit score been 20 points higher, the loan would have been approved,” clarifying why a rejection occurred and what historical aspects influenced it. The goal here is not to optimize future outcomes but to understand why something happened and identify the factors that influenced past decisions.

Despite these advancements, the studies mentioned above do not address the challenges of triggering interventions under uncertainty and resource constraints, which can significantly impact decision-making and lead to suboptimal policies, which we aim to address in this thesis. For example, the works in [86, 199] tackle different problems, while the work in [61] focuses on estimating causal effects, which aligns with a subset of the problem discussed in this thesis. Besides, we consider the work presented in [341] as a baseline for comparison, adapting it to a resource-constrained environment.

3.4. Reinforcement Learning in Prescriptive Process Monitoring

RL algorithms have gained significant attention for developing intervention policies to trigger actions during process execution, aiming to avoid negative outcomes or improve specific key performance indicators (KPIs). For instance, the work presented in [164] develops a policy via a Q-learning algorithm that recommends the next best activity for a target actor to enhance a particular KPI. Furthermore, the work in [86] compares policies based on causal inference with those based on RL algorithms, providing a quantitative comparative analysis of process outcome optimization problems using both approaches.

However, the above RL-based PrPM approaches use value-based algorithms that focus on learning the value function, deriving the policy indirectly by selecting actions that maximize the value function. Such methods need to estimate the value of each action and thus can struggle with large or continuous action spaces. Also, it relies on epsilon-greed for exploration strategies, which can be less efficient and may require careful tuning to balance exploration and exploitation [158].

In other RL PrPM approaches, *policy-based* algorithms are employed to learn intervention policies directly. These algorithms incorporate exploration via stochastic policies, enabling continual exploration of action possibilities to discover optimal policies. For instance, the work in [59] utilizes the *Proximal Policy Optimization (PPO)* [165, 166] algorithm to trigger proactive actions during process execution. It determines when to adapt during run-time based on predictions of outcomes and their reliability. However, this approach does not explicitly define the interventions to be triggered and does not estimate their effectiveness. Building on this, subsequent work in [341] extends the policy-based RL approach by explicitly defining an intervention and integrating its effects into the policy learning process. In addition, outcome predictions and conformalized prediction sets are used to guide the agent towards more certain intervention decisions. Thus, we consider the works in [341] and [59] as baseline comparisons for the approaches presented in this thesis.

Nevertheless, the RL-based PrPM approaches discussed above do not address the issue of who will execute the intervention. These approaches assume unlimited resources and the ability to trigger interventions immediately for all cases. For instance, in a scenario where a customer needs a phone call to offer a discount, the assumption is that a worker will always be available to make the call. In practice, resources are limited, and neither all cases can be intervened nor many of them. Addressing these limitations forms a key focus of this thesis.

3.5. Summary

This chapter reviewed state-of-the-art techniques in prescriptive process monitoring (PrPM), organizing them based on three main criteria identified in the literature: optimization objectives, intervention types, and policy learning approaches. PrPM techniques were categorized according to their optimization objective, either improving process outcomes or enhancing performance metrics. Intervention types were classified as either discrete or continuous, depending on the nature of the actions taken to achieve these objectives. Simultaneously, policy learning approaches were distinguished as either white-box (rule-based) or black-box (data-driven). This categorization helped to refine the scope of this thesis, which focuses on optimizing binary and categorical process outcomes using a single type of discrete intervention.

In line with the scope of this thesis, this chapter highlights two key limitations in existing work: the assumption of unlimited resources for executing interventions and the lack of consideration for uncertainty in intervention decisions. To contextualize our contributions, Table 5 provides a comparative summary, distinguishing our work from closely related studies within the thesis scope. It compares outcome-oriented PrPM approaches, emphasizing key criteria such as resource constraints, policy learning types, intervention importance/urgency, and uncertainty modeling. While existing works (e.g., [46], [47]) focus on white-box or black-box policy learning with limited uncertainty awareness, and others (e.g., [341], [86]) incorporate causal effects, none simultaneously address resource limitations or holistically quantify uncertainty across all intervention facets.

Table 5. Comparison of outcome-oriented prescriptive process monitoring approaches.

Approach	**Capacity**	**Policy Learning Type**	**Intervention Importance**		**Intervention Urgency**	**Measure of Uncertainty**		
	(Resource Constraint)	**(White/Black-Box)**	**Need (Outcome Prediction)**	**Effect (Causal Effect)**	**Intervention Window**	**Need**	**Effect**	**Urgency**
[46]	No	White-box	Yes	No	No	No	No	No
[47]	No	White-box	Yes	No	No	No	No	No
[59]	No	Black-box	Yes	No	No	Yes	No	No
[341]	No	Black-box	Yes	Yes	No	Yes	No	No
[86]	No	Black-box	No	Yes	No	No	No	No
Ours	Yes	Both	Yes	Yes	Yes	Yes	Yes	Yes

This comparison highlights the importance of incorporating critical aspects such as the importance, urgency, and uncertainty of interventions, which are vital for developing effective intervention policies. Such policies can address three core questions: who needs the intervention, when it should be triggered, and who should execute it. Building on prior research, this thesis advances the field by incorporating resource constraints, dual policy learning types, and explicit uncertainty modeling, thereby enhancing decision-making in interventions.

The following chapters will present the main contributions of this thesis, introducing PrPM approaches that integrate uncertainty awareness and resource constraints, designed to operate in both white-box and black-box settings, offering different solutions for optimizing intervention decisions in PrPM.

4. UNCERTAINTY-AWARE PRESCRIPTIVE PROCESS MONITORING

This chapter presents an uncertainty-aware prescriptive process monitoring approach, hereafter referred to as *UN-PrPM*. The *UN-PrPM* aims to develop an intervention policy that identifies which instances (or cases) within a business process require intervention to enhance process outcomes and maximize a *total gain* function. This approach establishes a baseline that will be expanded upon in subsequent chapters.

Like existing PrPM approaches, the *UN-PrPM* operates under resource-unaware and resource-unconstrained conditions, where interventions are triggered as needed for all cases without regard to resource limitations [47, 46]. However, a notable limitation of current PrPM approaches is their neglect of the inherent uncertainty in factors influencing the intervention policy, which can result in inefficient decision-making. For instance, PrPM techniques often rely solely on predictive process monitoring predictions to determine which cases require intervention, without considering the uncertainty in these predictions. Accordingly, in this chapter, we address the following research question: **RQ1:** *How can we develop intervention policies that effectively incorporate uncertainty into decision-making, ensuring that predictions are reliable and decisions are made confidently to maximize a total gain function?*

To address this question, *UN-PrPM* introduces a key distinction: it incorporates uncertainty awareness into the intervention decision-making process, setting it apart from conventional resource-unconstrained PrPM approaches. In *UN-PrPM* , the intervention policy is based on estimates from a predictive model; specifically, it employs a predictive-based policy that triggers interventions based on these estimates. For instance, similar to previous work by [55, 54], an intervention might be triggered when the probability of a negative outcome exceeds a predefined threshold (e.g., 0.5). However, *UN-PrPM* goes a step further by explicitly defining the intervention policy and integrating prediction uncertainties, thereby enhancing the reliability of intervention decisions.

To achieve this, we first outline the pipeline for PPM models and provide an empirical evaluation to assess the performance of various PPM techniques with different sequence encoding methods. Based on the evaluation results, we identify the best PPM techniques in terms of machine learning algorithms and sequence encoding techniques, and we then explore methods for estimating uncertainties associated with predictions, specifically uncertainty quantification and conformal prediction. Finally, we discuss how these predictions, along with their uncertainties, are integrated into the operationalization of *UN-PrPM*.

This chapter covers the PPM pipeline (Section 4.1), PPM evaluations (Section 4.2), and uncertainty methods (Section 4.3). It then presents *UN-PrPM* (Section 4.4), its evaluation (Section 4.5), and concludes with a summary (Section 4.6).

4.1. Predictive Process Monitoring Pipeline

The process mining literature widely discusses the predictive process monitoring (PPM) pipeline [99, 10, 18]. To ensure this thesis is self-contained, we include a general overview of the PPM pipeline. The pipeline can be divided into two main phases: the *offline* phase and the *online* phase.

4.1.1. Offline Phase of PPM

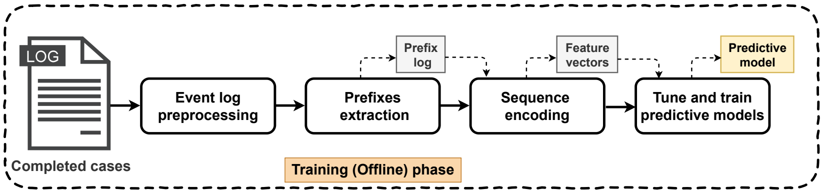


Figure 5. Predictive process monitoring pipeline (offline phase).

The offline phase, shown in Fig 5, involves preparing and training a predictive model based on completed historical cases. The key components of this phase are *event log preprocessing*, *prefixes extraction*, *sequence encoding*, and *model tuning and training*. The output of the offline phase is a trained *predictive model* that is ready for use.

Event log preprocessing. This component helps to ensure accurate predictions and intervention decisions in predictive and prescriptive tasks. It includes *data cleaning* and *feature engineering* to guide an ML algorithm's learning process.

The *data cleaning* step typically includes removing incomplete cases and correcting data types and formatting, as suggested by [10]. Incomplete cases are those where the final activity, representing the outcome, is not observed. These cases should be removed from the original log to ensure the historical cases accurately represent actual process execution. At the same time, correcting data types ensures that all data types are correct and consistently formatted. It includes, for instance, correcting incorrect timestamps or ages.

In the *feature engineering* step, we distinguish between two types of features in process mining: *intra-case* and *inter-case* features. *Intra-case* features represent individual business cases or are defined for each case in isolation [167]. In contrast, *inter-case* features represent concurrent cases executed simultaneously or computed across all ongoing cases [168]. In this thesis, we enriched event logs with intra-case and inter-case features to provide ML algorithms with as much information as possible to ensure accurate predictions, as outlined in [174, 53].

Intra-case features can be *temporal* or *domain-specific*, capturing details relevant to each individual case. *Temporal intra-case* features, as suggested in previous work [10, 18], may include: "time to the last event," time since case start," time since last event," time since midnight," month," "weekday," and "hour." In addition to these temporal features, *domain-specific* features can also be included.

For example, in the loan origination log introduced later in the thesis, we include features to encode changes in key features such as changes in the number of loan offers sent to applicants, changes in the monthly interest rate offered to the applicant, and changes in the number of terms of the loan.

Moreover, the study of *Inter-case* features, which identify relationships between different cases, has gained notable attention [314]. However, their application often remains domain-specific, limiting broader adoption by the research community. In this context, our work focuses primarily on capturing *workload* and *demand intensity*, according to Little's Law [169]. For example, *Work in Progress (WIP)*, the number of active cases, and *arrival rate (λ)*, i.e., the number of cases created per time unit. Consequently, we include features corresponding to these measures. Additionally, we include features that capture the number of past events within a specific time window, such as the last hour. These features provide a broader context of the process environment, allowing the predictive model to account for the workload, the intensity of demand, and the temporal dynamics that may influence the outcome of individual cases.

The event log preprocessing component output is a prepared and enriched event log. This log contains complete cases with features that were either originally present or manually engineered during the preprocessing phase. These features include both intra-case and inter-case attributes, providing a dataset ready for use in the subsequent stages of the predictive monitoring pipeline.

Prefixes extraction. The *prefixes extraction* component simulates real-life situations in PPM. In practice, a predictive monitoring approach aims to make predictions for ongoing cases instead of completed cases. To simulate this scenario, we use the prepared and enriched event log to extract all possible prefixes for each case up to a certain length, k, via a prefix function, i.e. $P(\sigma, k)$ (see Def. 2.5.1). For example, for a complete trace consisting of a total of 10 events, we can consider up to 9 prefixes: the incomplete trace after executing the first event, the incomplete trace after executing the first and second events, and so on. This method allows us to capture the dynamic nature of ongoing cases and provides a robust dataset for training predictive models that can effectively handle real-time data.

However, using all possible prefixes has several drawbacks. Firstly, the large number of all possible prefixes to be extracted can affect the training process of predictive models and slow it down. Secondly, if case lengths are extremely different, a potential bias may occur due to the predominance of longer cases. Therefore, according to the literature, it is common to set the length of possible prefixes up to a certain maximum, such as the minimum between 20 or the 90th percentile of positive cases [10, 18, 170, 9]. Positive cases usually represent the process working as intended and provide a clearer signal for the model, reducing the impact of noise from negative cases. This approach helps maintain a manageable dataset size, reduce computational overhead, and mitigate bias towards longer cases, ensuring efficient training of predictive models. Consequently, the prefix extraction step output is a *prefix log* containing all case prefixes up to the k length.

Table 6. Encoding techniques adapted from [10, 18].

Encoding technique	Case abstraction	Categorical	Numeric
Static	All case attributes except for the Case ID	One-hot	As is
Last-State	Last event attributes	One-hot	As is
Aggregate	All events, unordered	Frequencies	Sum, min, max, mean, std
Index	All events, ordered	One-hot for each index	As is for each index

Sequence encoding. This component transforms each case prefix into a fixed-size feature vector using a sequence encoder function, i.e., $enc(P(\sigma,k))$ (see Def. 2.5.2). This transformation is necessary for training ML algorithms. As discussed in Section 2.5.1, the literature outlines several sequence encoding techniques, including *static* encoding for case attributes and methods such as *last-state*, *aggregate*, *index*, and c*ombined encoding* for event attributes.

Table 6 summarizes these encoding techniques. For instance, case attributes are mapped to encoded features based on their data type. Categorical, such as "loan type," are encoded using methods like one-hot encoding, while numerical data types are left as is. For event attributes, and based on an aggregate encoding technique depends on the data type: categorical data is encoded based on frequency, while numeric data is encoded using aggregate functions such as sum, min, or max.

In this thesis, we evaluate these encoding techniques to identify which yields the most accurate predictions. This evaluation is important because effective intervention decisions rely on precise predictions, particularly when resource allocation is limited.

Model tuning and training. The main goal of this component is to train and tune (or *optimize*) ML algorithms to predict process outcomes for unseen or ongoing cases. Predicting outcomes for previously seen cases is trivial, as the models could simply memorize these cases and retrieve the associated outcomes. Therefore, training and tuning the model to generalize to new, unseen cases is necessary. This ensures that the model can accurately predict outcomes for ongoing cases rather than memorize past cases.

Tuning ML algorithms involves adjusting various *hyperparameters*, which are parameters the analyst sets during the training process to control the model's complexity. Each ML algorithm has different hyperparameters that need to be optimized (see discussion in Section 2.1). This tuning step is important to avoid *overfitting* and *underfitting*. *Overfitting* occurs when the model is too complex (or more hyperparameters to be optimized) and captures the noise in the training data, failing to generalize to new cases. *Underfitting* occurs when the model is too simple (or fewer hyperparameters to be optimized) and cannot capture the underlying patterns in the data. Both overfitting and underfitting lead to low prediction accuracy and, thus, incorrect intervention decisions, making model tuning necessary for developing accurate predictive models.

A common approach to tuning ML algorithms is *n-fold cross-validation* [171]. This method involves splitting the logs into *training* and *testing* sets. The training set is then partitioned into n equally sized folds. The model is trained n times, each time using $n-1$ folds for *training* and the remaining fold for *validation*. This process is repeated n times, with each fold serving as the validation set exactly once. The performance metrics are averaged over the n iterations to evaluate the model's performance. This approach avoids partitioning data into independent subsets, thereby maximizing the number of training cases and enhancing the model's performance.

Similarly, it is common to use sequential or adaptive hyperparameter optimization techniques to choose the best hyperparameter configurations that result from the tuning process and use them in the final training for ML algorithms. Sequential techniques, such as the Tree-structured Parzen Estimator (TPE) [172, 173], require analysts to specify only the ranges and sampling distribution for each hyperparameter. This approach is similar to random search but differs in that sequential optimization techniques perform iterative optimization. In each iteration, a promising hyperparameter configuration is chosen based on previously tried settings.

Once the best hyperparameters are chosen, the training and validation sets are combined to form a final training set, which is used to train the ML algorithm with the optimized hyperparameters. The trained model is then evaluated using the testing set to assess its generalizability to unseen cases. The output from this tuning and training component is a trained predictive model ready for use during operation to monitor ongoing cases and predict process outcomes.

4.1.2. Online Phase of PPM

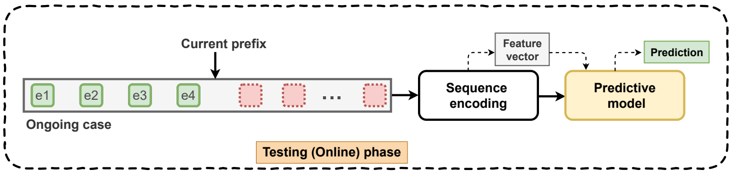


Figure 6. Predictive process monitoring pipeline (online phase).

In the online phase (Fig. 6), the PPM approach monitors ongoing cases and makes predictions. The trained predictive model is applied to ongoing cases at various prefixes. An ongoing case at a given prefix is encoded using a sequence encoder into a fixed-size feature vector. The trained predictive model then makes predictions. The output is an *outcome prediction (OP)*, representing the probability of a negative outcome. For example, in a loan origination process, this prediction indicates the probability of canceling the loan application or rejecting the loan offer. If the predicted probability exceeds a certain threshold (e.g., 0.5), the case at that prefix is classified as negative, necessitating proactive intervention to prevent a costly negative outcome.

4.2. Predictive Process Monitoring Evaluation

This section presents comparative analytics for different PPM approaches by varying sequence encoding techniques and ML algorithms, which are discussed in Section 2.5. The goal is to select the best combination of sequence encoders and ML algorithms for predicting ongoing case outcomes that will be used later in a prescriptive monitoring setting. To structure and guide this evaluation, we subdivide RQ1 into the following sub-research questions:

RQ1.1: How do different PPM approaches compare in terms of prediction accuracy, and which approach performs the best?

RQ1.2: To what extent do inter-case features influence the performance of predictive process monitoring approaches?

In the following subsections, we first present the evaluation datasets, then explain the evaluation strategy, and end with a discussion of the experimental results.

4.2.1. Datasets

The selection of event logs was constrained by the fact that PrPM approaches require event logs in which: (i) there is a concept of a "*case outcome*," and the outcome is recorded for each completed case; (ii) there is an "*intervention*" (an action intended to increase the likelihood of a positive outcome) and instances of this intervention are recorded in the dataset. Accordingly, we restricted the evaluation to event logs with the associated description or documentation referencing a case outcome and an intervention.

This thesis utilizes publicly available real-life event logs[1], particularly *BPIC2017*[2] and *BPIC2012*[3], which represent a loan origination process where an applicant submits a loan application that undergoes several steps. Initially, the application is reviewed for completeness by a process worker, such as a loan officer, who then notifies the applicants about the acceptance of their application. The process worker proceeds to process the application, prepares a suitable offer, and sends it to the applicant. The applicant can either accept or decline the offer directly, triggering the loan officer to prepare another suitable offer.

Table 7. Summary of event logs characteristics

Log	# Cases	# Events	Mean length	Last activity	Outcome	Intervention activity
BPIC2012	12,688	156,962	12	A_Approved	Positive	-
				A_Canceled A_Declined	Negative	Send_Offer
BPIC2017	31,411	1,198,319	38	A_Pending	Positive	-
				A_Canceled A_Declined	Negative	Send_Offer

---

[1] `https://www.tf-pm.org/resources/xes-standard/about-xes/event-logs`

[2] `https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b`

[3] `https://data.4tu.nl/articles/dataset/BPI_Challenge_2012/12689204/1`

The outcome of a case is determined by a condition (boolean function) evaluated on a completed case, as detailed in Table 7. The process concludes with a positive outcome if the applicant accepts the bank's offer and signs the contract. For instance, in BPIC2017, we label completed cases ending with the activity "A_Pending" as having a positive outcome. Conversely, the process ends with a negative outcome if the applicant rejects the loan offer or the bank cancels the application due to, for instance, document incompleteness. Thus, completed cases ending with either "A_Canceled" or "A_Declined" are labeled with a negative outcome. Both logs exhibit class imbalance, as shown in Fig. 7: BPIC2017 has a near-balanced ratio (the proportion of majority class cases) of 1.2 (minority class $\approx 45\%$ of cases), indicating mild imbalance, while BPIC2012 shows a 4.65 ratio (minority class $\approx 17.7\%$), reflecting moderate imbalance.

On the other hand, *the intervention*, i.e., sending another loan offer, is triggered in a way to affect cases ending with a negative outcome positively. This means that the intervention can transform negative cases into positive ones only if executed; otherwise, cases will continue to end with a negative outcome. Reports from previous analyses of the loan origination process have shown that the number of cases ending with a positive outcome increases by 10% when receiving more than one loan offer [175]. Therefore, in this thesis, we choose to use sending an additional loan offer as the intervention to improve the chances of achieving a positive outcome. As shown in Fig. 7, in BPIC2017, 71% of the cases are in the control group (i.e., received only one offer), while 29% receive treatment (i.e., received more than one offer). In BPIC2012, the distribution is 64% control and 36% treatment, reflecting a moderate skew.

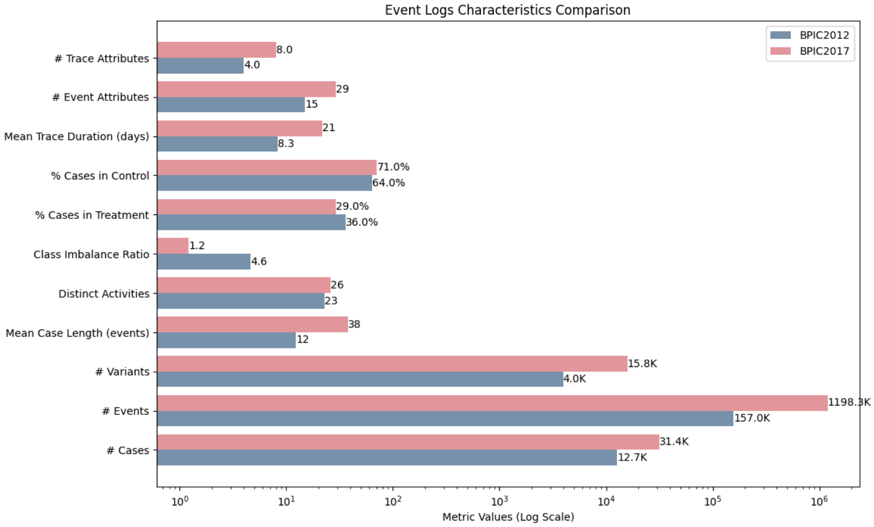


Figure 7. Event logs characteristics comparison.

The event logs also exhibit distinct structural and behavioral characteristics, as shown in Table 7 and Fig. 7, making them well-suited for comparative analysis. BPIC2017 has 26 distinct activities, slightly more than BPIC2012's 23, indicating greater control-flow complexity. However, BPIC2017 cases have a mean trace duration of 21 days, nearly three times longer than BPIC2012's 8.3 days, reflecting differences in process execution. Case lengths also vary: BPIC2017 averages 38 events per case, suggesting more granular workflows, while BPIC2012 averages 12. Further, BPIC2017 provides richer metadata, with 29 event-level and 8 trace-level attributes, compared to BPIC2012's 15 and 4, enabling more detailed contextual analysis that would affect the performance of intervention policies. We also examined the Pareto/non-Pareto distribution of control-flow variants (unique activity sequences) in both logs. Both exhibit a non-Pareto distribution, meaning no small subset of variants dominates the process. Specifically, the top 20% of control-flow variants account for less than 80% of cases, indicating high variability in how activities are sequenced.

4.2.2. Experimental setup

This section outlines the evaluation measures used to compare different configurations for predictive monitoring. We also describe the methodology for splitting the logs to align with the business process context and optimizing the hyperparameters of the ML algorithms under comparison. All experiments are conducted using Python 3.10 and the scikit-learn library [176] on a virtual machine with an Intel Xeon Processor (Skylake, IBRS) @ 2.10 GHz and 64GB of RAM.

We follow a temporal split approach to simulate real-life scenarios where predictive models are trained on historical cases and then used to monitor and make predictions for ongoing cases. This involves splitting the event logs into training (80%) and testing (20%) sets based on timestamps in chronological order. The first 80% of the cases are used for optimizing ML algorithms' hyperparameters, selecting the best hyperparameters, and training the final model. The remaining 20% of the cases are used during operation time to evaluate the performance of the final model.

Performance Measures. We evaluate and compare PPM approaches using the AUC as the primary performance metric. The AUC is chosen for its threshold-independent nature and ability to provide an unbiased measure of model performance, especially in highly imbalanced class labels. Additionally, we report the F1-scores computed at the default threshold of 0.5 to address potential biases arising from relying on a single metric. The objective is to identify the approach that achieves the highest AUC, which will be considered the best-performing model for subsequent prescriptive monitoring settings.

Moreover, we utilize *SHAP* (SHapley Additive exPlanations) to examine the impact of inter-case features, e.g., work in progress, the number of past events, and arrival rate, on the model's predictions [177]. It explains the contributions of various features to the model predictions, indicating how each feature influences prediction accuracy. This analysis is important for understanding the extent to which inter-case features affect the accuracy of predictive models, thereby ensuring precise intervention decisions and effective resource allocation.

Sequence Encoders and Classifiers Learning. We experimented with four sequence encoders to encode event attributes—last state, aggregate, index, and combined. These encoding techniques are combined with the static encoding technique to encode static attributes (see discussion in Section 2.5.2). By default, the static encoding technique is employed first, followed by another technique to encode event attributes. We refer to this combination by the name of the event attributes encoding technique; for example, static plus last state is referred to as the last-state encoding technique, and so on. These encoding techniques have shown promising results in various predictive monitoring tasks [18, 10, 9, 14, 15]. We aim to compare these techniques to identify the one that produces the highest accuracy, ensuring reliable predictions for subsequent prescriptive tasks.

Additionally, we experimented with four ML algorithms to train and build predictive models: Categorical Boosting (CatBoost), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), and Random Forest (RF). Gradient boosted trees and RF algorithms are included due to their superior performance compared to other classification algorithms in empirical studies across 165 datasets [178, 179]. RF is relatively simple to implement with minimal hyperparameter tuning and balances bias and variance. Also, the RF has been shown to outperform many other methods (such as decision trees) in both predictive monitoring [167, 180] and more general empirical investigations [181].

Table 8. Hyperparameters and distributions used in optimization via TPE.

Classifier	Hyperparameter	Distribution	Values
CatBoost	Learning Rate	Uniform	$0.0 \leq x \leq 1.0$
	Subsample	Uniform	$0.05 \leq x \leq 1$
	Max Depth	Quantized Uniform	$1 \leq x \leq 16$, step=1
	L2 Leaf Reg	Log Uniform	$1 \leq x \leq 16$
	Number of Estimators	Choice	$[100, 250, 500, 1000]$
XGBoost	Learning Rate	Uniform	$0.0 \leq x \leq 1.0$
	Max Depth	Quantized Uniform	$2 \leq x \leq 16$, step=1
	Min Child Weight	Quantized Uniform	$1 \leq x \leq 6$, step=1
	Subsample	Uniform	$0.5 \leq x \leq 1$
	Colsample By Tree	Uniform	$0.5 \leq x \leq 1$
	Number of Estimators	Choice	$[100, 250, 500, 1000]$
LightGBM	Learning Rate	Uniform	$0.0 \leq x \leq 1.0$
	Max Depth	Quantized Uniform	$2 \leq x \leq 16$, step=1
	Min Child Weight	Quantized Uniform	$1 \leq x \leq 6$, step=1
	Subsample	Uniform	$0.5 \leq x \leq 1$
	Colsample By Tree	Uniform	$0.5 \leq x \leq 1$
	Number of Estimators	Choice	$[100, 250, 500, 1000]$
RF	Number of Estimators	Choice	$[100, 250, 500, 1000]$
	Max Depth	Quantized Uniform	$2 \leq x \leq 16$, step=1
	Min Samples Split	Quantized Uniform	$2 \leq x \leq 20$, step=1
	Min Samples Leaf	Quantized Uniform	$1 \leq x \leq 16$, step=1
	Max Features	Choice	$[auto, sqrt, log2, None]$

On the other hand, we include gradient boosting algorithms since they have recently gained more attention and demonstrated good results when applied to business process event logs [174, 87]. For example, CatBoost is designed to handle categorical features without extensive preprocessing, such as one-hot encoding, which can significantly simplify the data preparation process and improve model performance. However, CatBoost can be computationally intensive and require more resources and time to train compared to simpler algorithms. Similarly, XGBoost and LightGBM are known for their performance and accuracy in various ML tasks [178], including PPM. However, they can be sensitive to their hyperparameters and may require careful tuning to achieve optimal performance, which can be challenging in hyperparameter optimization.

The classification algorithms' hyperparameters must be tuned to build optimized predictive models. Thus, we optimize the hyperparameters using the Tree-structured Parzen Estimator (TPE) algorithm for each combination: a dataset, a sequence encoding technique, and a classification algorithm. For each combination of hyperparameters (or a configuration), we conduct 3-fold cross-validation. During this process, the cases in the training set are randomly divided into groups, ensuring that any two prefixes belonging to the same case remain in the same group. Finally, we select the best configuration with the highest mean AUC calculated across the three folds. Table 8 shows the values and sampling distribution for each hyperparameter, which is given as input to the TPE optimizer.

4.2.3. Results

Here, we show comparative results between different combinations of a dataset, a sequence encoding technique, and a classification algorithm in terms of AUC and F1-scores with and without the inclusion of inter-case features. Then, we explore whether inter-case features contribute to model predictions using SHAP.

Table 9 shows the results of AUC and F1-Scores for both logs, with and without the inclusion of inter-case features to address RQ1.1 and RQ1.2. For the BPIC2012 dataset, the experimental results highlight that CatBoost with index encoding achieved the highest performance scores regarding the AUC and F1-score when inter-case features were included. The AUC score for this configuration reached 0.8804, notably higher than other methods, demonstrating the significant benefit of integrating inter-case features for this model. RF, XGBoost, and LightGBM also performed well, with each classifier showing improved AUC scores when inter-case features were incorporated. Among these, CatBoost and XGBoost showed the most consistent and high-performing results across different encoding techniques, making them strong candidates for further analysis. The F1-scores for these models remain relatively stable, indicating that including inter-case features primarily enhances the model's ability to distinguish between positive and negative outcomes without sacrificing precision or recall.

Table 9. Overall AUC (F1-score)

Dataset Name	Classifier	Encoding Technique	AUC Score (F1-score)	
			With Inter-case	Without Inter-case
BPIC2012	catboost	Combined	0.8394 (0.6726)	0.7922 (0.6726)
		Last-state	0.8334 (0.6728)	0.7601 (0.6728)
		Aggregate	0.7391 (0.5972)	0.7118 (0.5972)
		Index	**0.8804** (0.6882)	**0.8468** (0.6682)
	randomforest	Combined	0.7324 (0.5888)	0.7480 (0.5888)
		Last-state	0.7511 (0.6095)	0.7376 (0.6095)
		Aggregate	0.7403 (0.5983)	0.7351 (0.5983)
		Index	**0.8199** (0.6585)	**0.7772** (0.6585)
	xgboost	Combined	**0.8755** (0.6835)	0.7754 (0.6835)
		Last-state	0.7684 (0.6273)	0.7996 (0.6273)
		Aggregate	0.7743 (0.6331)	**0.8150** (0.6331)
		Index	0.8460 (0.6714)	0.8148 (0.6714)
	lightgbm	Combined	0.7831 (0.6414)	0.7949 (0.6414)
		Last-state	0.7989 (0.653)	0.7856 (0.653)
		Aggregate	0.8386 (0.6778)	0.7548 (0.6778)
		Index	**0.8393** (0.6752)	**0.8024** (0.6752)
BPIC2017	catboost	Combined	0.69048 (0.6141)	0.67052 (0.5141)
		Last-state	0.7145 (0.6507)	0.6945 (0.5007)
		Aggregate	0.7245 (0.6804)	0.6544 (0.5804)
		Index	**0.7439** (0.6247)	**0.7042** (0.6047)
	randomforest	Combined	0.6645 (0.5515)	**0.6045** (0.5017)
		Last-state	0.6718 (0.6533)	0.6644 (0.5511)
		Aggregate	**0.7056** (0.4908)	**0.6901** (0.5503)
		Index	0.6345 (0.5009)	0.6042 (0.4009)
	xgboost	Combined	**0.7045** (0.6215)	**0.7044** (0.5515)
		Last-state	0.6301 (0.5607)	0.6045 (0.4901)
		Aggregate	0.6740 (0.5304)	0.6208 (0.6303)
		Index	0.7020 (0.655)	0.6539 (0.607)
	lightgbm	Combined	**0.72032** (0.6007)	0.68045 (0.5607)
		Last-state	0.6744 (0.501)	0.6534 (0.491)
		Aggregate	0.7112 (0.637)	**0.6445** (0.507)
		Index	0.7125 (0.6088)	0.6536 (0.5588)

The BPIC2017 dataset presented less consistent results. CatBoost with index encoding again delivered the highest AUC score of 0.7439, indicating the value of inter-case features in enhancing model performance. However, the overall performance metrics were lower compared to BPIC2012, suggesting that the BPIC2017 dataset has more challenges for predictive modeling. RF, XGBoost, and LightGBM exhibited varied results, with improvements in AUC when inter-case features were included but with less significant changes than seen in BPIC2012. Notably, LightGBM and XGBoost also demonstrated improvements in AUC and F1-Scores with inter-case features, though the impact was not high. The F1-Scores for BPIC2017 are also generally lower than those for BPIC2012, suggesting that the classification task for BPIC2017 may be inherently more challenging.

CatBoost with *index encoding* outperforms other combinations due to the strengths of the CatBoost algorithm and the advantages of the index encoding technique. The combination of CatBoost with index encoding consistently performed well across both event logs, showing the highest AUC scores and demonstrating substantial benefits from including inter-case features. This is due to the strengths of the CatBoost algorithm and the advantages of the index encoding technique.

CatBoost is designed to handle categorical features without extensive preprocessing, preserving the inherent structure and relationships within the data. It also uses ordered boosting, which processes the data to reduce overfitting, a feature particularly useful for datasets with high dimensionality, such as event logs. The index encoding technique captures the position of events within a case, which is important for predictive tasks where the order of events impacts the outcome. Thus, it retains the temporal sequence, helping the model learn patterns over time.

Accordingly, based on our findings, we decided to use *CatBoost with index encoding* to develop predictive models for intervention policies that rely on outcome predictions, i.e., triggering interventions when the outcome prediction exceeds a certain threshold.

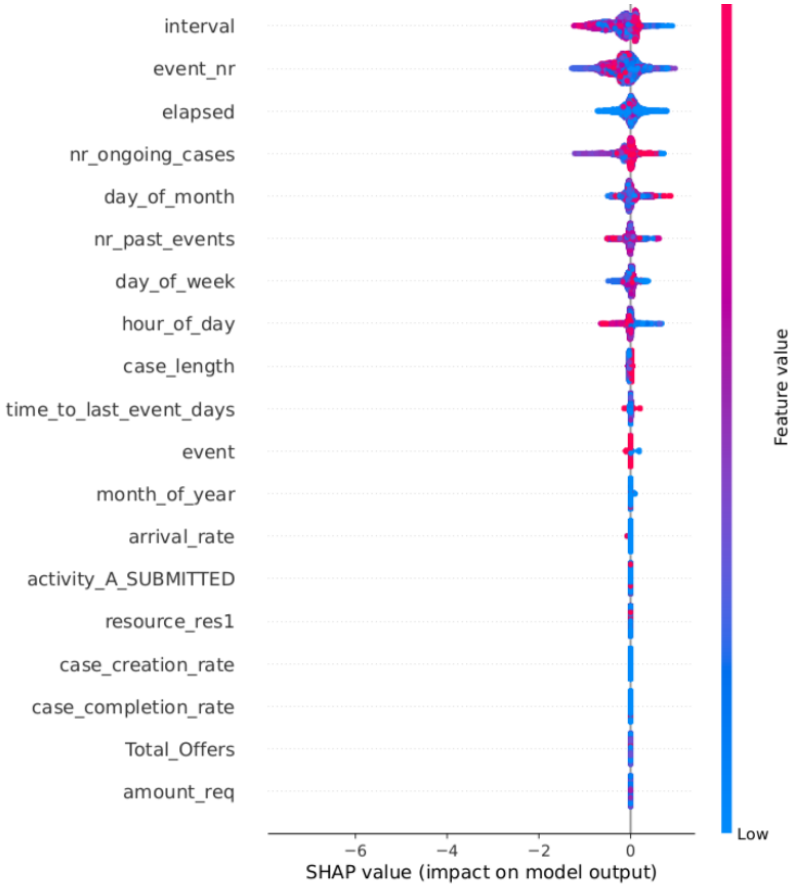


Figure 8. An overview of which features are most important for a predictive model.

To further understand the contribution of inter-case features to the model's predictions, we conducted a SHAP analysis, as shown in Fig. 8. The x-axis represents SHAP values, which indicate the impact of each feature on the model's output. Positive SHAP values push the prediction towards a positive outcome (higher prediction), while negative SHAP values push it towards a negative outcome (lower prediction). The y-axis lists the features that most influence model predictions, sorted from top to bottom. The analysis shows that inter-case features, such as the number of ongoing cases (WIP), arrival rate, and the number of past events, influence the model's predictions. For instance, WIP has a wide spread of SHAP values, indicating its significant positive and negative impacts on the model's predictions. Additionally, temporal features such as the day of the month and domain-dependent features like the total number of offers sent to the applicant also significantly impact predictions.

The above results emphasize the importance of integrating inter-case features, reinforcing the decision to use CatBoost with index encoding to develop accurate predictive models. These models generate accurate outcome predictions, i.e., the probability of a negative outcome. These outcome predictions are used to develop an intervention policy that triggers interventions when they exceed a certain threshold. However, the quality of these outcome prediction scores may inherently come with a particular level of uncertainty that affects intervention decisions, potentially leading to non-optimal intervention policies. Accordingly, in the following section, we describe how to estimate uncertainties associated with predictive model predictions.

4.3. Uncertainty Estimation

Developing intervention policies based only on low-quality outcome prediction scores can be misleading. For instance, consider a predictive model that assigns a probability of 0.7 to a case, indicating it is more likely to end with a negative outcome. An intervention policy based only on that prediction would trigger an intervention to prevent the negative outcome. However, if that prediction score comes with a high level of uncertainty, it could lead to incorrect intervention decisions, wasting limited resources. Therefore, integrating outcome prediction scores with a measure of the associated uncertainty is necessary for developing an uncertainty-aware PrPM approach, referred to as *UN-PrPM*.

In this thesis, we use two methods to measure the level of uncertainty: *uncertainty quantification* and *conformal prediction*, as discussed in Section 2.1.2.

4.3.1. Uncertainty quantification

In this thesis, we follow the approach provided by [37] to quantify the total uncertainty associated with outcome prediction. This involves constructing a predictive model using ensemble learning, as depicted in Fig. 9. The primary principle of ensemble learning is to build one strong predictive model from multiple weak models. This approach ensures the improvement of overall prediction performance, reduces the risk of overfitting, and avoids local minima. Hence, ensemble learning has two purposes. First, it ensures accurate prediction of the probability of a negative outcome, i.e., *outcome prediction (OP)*. Second, it provides a means to estimate the *total uncertainty (TU)* associated with this prediction.

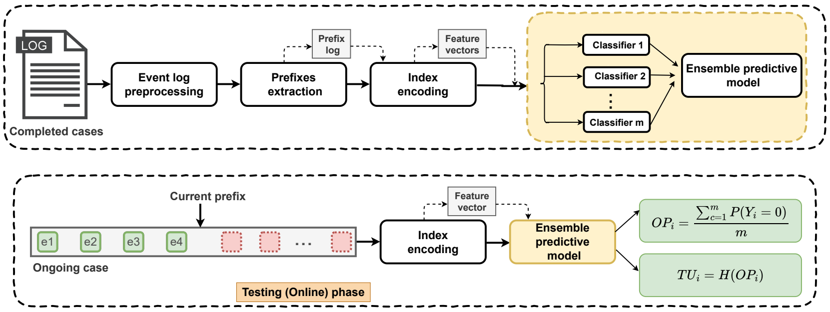


Figure 9. An overview of the ensemble predictive model to obtain outcome (OP) predictions and quantify the total uncertainty (TU).

A single classifier in the ensemble is a probabilistic model that takes a feature vector as input and outputs the outcome prediction, i.e., the probability of a negative outcome. In the ensemble, we have multiple classifiers, ranging from 1 to mmm, where mmm is the number of classifiers. The outcome prediction from the ensemble is determined by aggregating scores derived from an ensemble of predictions, as shown in Eq. 4.1. Considering a single classifier from the ensemble, Y_i represents the outcome of a case i, where Y_i is binary (0 for negative outcome, 1 for positive outcome). Hence, OP_i is the aggregated outcome prediction for the case i from the ensemble model. This aggregation allows us to capture the variability of different classifiers and quantify the total uncertainty of the OP.

$$OP_i = \frac{\sum_{c=1}^{m} P(Y_i = 0)}{m} \tag{4.1}$$

The *total uncertainty (TU)* is quantified based on the *entropy (H)*, i.e., describing the average level of surprise or uncertainty inherent in the possible outcome of the aggregated outcome prediction from the ensemble, see Eq. 4.2. The TU is a value within the range of 0 to 1, where values closer to 0 indicate higher certainty in the outcome prediction, while values closer to 1 signify greater uncertainty.

$$TU_i = H(OP_i) \tag{4.2}$$

To exemplify the estimation of total uncertainty (TU), consider the following scenarios: (1) If we give input that all classifiers understand and yield similar outcome prediction scores, classifiers are confident about their prediction scores, resulting in minimal total uncertainty. (2) If we show input that all classifiers understand and generate identical predictions but a high entropy distribution over outcomes, classifiers are uncertain with high data uncertainty. (3) If we present the ensemble with inputs that none of the classifiers understand, all classifiers yield different outcome prediction scores because the input is far from the training data. Consequently, the ensemble is highly diverse and has high entropy, indicating high knowledge uncertainty as we average various probability distributions.

Integrating both outcome prediction (OP) and total uncertainty (TU) guides triggering interventions at different decision points, providing information about the need for intervention. For instance, when the predictive model indicates a high probability of a negative outcome for a case but with high uncertainty, the decision might be against triggering the intervention for that particular case, thereby preserving resources and efforts. Instead, intervention could be triggered in another case where the predicted outcome carries a higher level of certainty. Thus, considering TU introduces varying degrees of importance among different cases, allowing for more informed intervention decisions.

4.3.2. Conformal Prediction

Conformal prediction is a framework that converts point estimates from any underlying predictive model, i.e., model-agnostic, into a *prediction set* for classification tasks or a *prediction interval* for regression tasks. Conformal prediction provides a rigorous way to estimate uncertainty, ensuring that the exact point estimate is contained within the prediction interval or set with a specified confidence level $(1-\alpha)$, thereby enhancing the reliability of intervention decisions. This is in contrast to total uncertainty estimation, which is based on a heuristic method to quantify uncertainty without providing any confidence level or guarantee of coverage. By having theoretical guarantees, conformal prediction adds more confidence in decision-making processes.

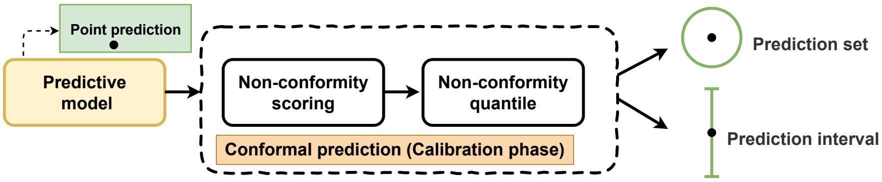


Figure 10. An overview of the conformal prediction framework.

Conformal prediction is considered a post-processing step and is performed during a *calibration phase*, as shown in Fig. 10, to convert outcome prediction (the point estimate: OP) into a conformalized outcome prediction (the prediction set: COP). Following calibration, we append the letter "C" to signify conformalized estimates. The COP set is shown in Eq. 4.3, where $(\hat{Y}_i, p_i)$ represents each prediction for a given case i, with $\hat{Y}_i$ being the predicted outcome and p_i the associated confidence level. Y_i denotes the set of possible outcomes, e.g., $Y_i = 0$ for the negative outcome, and α is the significance level controlling the error rate.

$$COP_i = \{(\hat{Y}_i, p_i) \mid \hat{Y}_i \in Y_i, p_i \geq 1-\alpha\} \tag{4.3}$$

This COP set can take four different forms in binary classification settings:

- (a) - {0,1}: The set includes both outcomes, indicating high uncertainty about the prediction.
- (b) - {0}: The set includes only the negative outcome, indicating certainty in predicting a negative outcome.
- (c) - {1}: The set includes only the positive outcome, indicating certainty in predicting a positive outcome.
- (d) - { }: The set is empty, indicating extreme uncertainty where no outcome meets the desired confidence level.

In prescriptive monitoring settings, the prediction set's most important form is the one with only the negative outcome, i.e., {0}. This is because the goal is to trigger interventions for cases we confidently know are more likely to end with a negative outcome. Cases with the prediction sets {0,1} or { } indicate higher uncertainty. By focusing on cases where the prediction set is {0}, we can prioritize interventions for those cases with high confidence in their negative outcome, thereby optimizing intervention decisions. Thus, The *COP*, as previously discussed, can take four forms, but for simplicity, it is encoded as either 0 or 1. A value of 1 represents a high need for intervening when the COP set includes only the negative outcome ($COP = \{0\}$), while a value of 0 reflects uncertainty about the negative outcome, indicating less need and representing the other three forms of the *COP*. For example, if the COP set contains only the negative outcome prediction ($Y = 0$), i.e., $COP = \{0\}$, the need is 1; otherwise, it is 0.

To construct the *COP*, assuming we have a predictive model and a calibration set independent of the training set, we first use the predictive model to generate outcome predictions for each instance of the calibration set. Then, a non-conformity scoring method generates non-conformal scores, such as one minus the outcome prediction of the true class. Higher non-conformity scores indicate higher uncertainty, meaning the instance in the calibration set does not conform to the training instances. Given the generated scores for each instance of the calibration set, a non-conformity quantile is defined as an empirical quantile of the non-conformity scores. Finally, given an unseen case, the *COP* is constructed as shown in Eq. 4.3, and it is guaranteed to contain the true outcome with probability $1-\alpha$ even if the underlying model is not accurate, as demonstrated in [182].

4.4. Prescriptive Process Monitoring Under Uncertainty

This section proposes an uncertainty-aware prescriptive process monitoring approach, *UN-PrPM*, as depicted in Fig. 11. This approach develops an intervention policy that triggers interventions for cases more likely to end with a negative outcome, incorporating confidence measures to enhance decision-making accuracy. We hypothesize that adding uncertainty measures makes the intervention policy more conservative, while achieving good performance by accurately targeting negative cases that benefit from the intervention. This contrasts with policies that rely solely on estimates with potentially high levels of inherent uncertainty. Therefore, the *UN-PrPM* approach integrates a predictive model with a method to measure the uncertainty associated with model predictions.

In this context, we assume that each case ending with a negative outcome incurs a cost, while each case ending with a positive outcome yields a gain. The objective is to maximize the number of cases that end with a positive outcome, thereby increasing the positive outcome gain and minimizing the negative outcome cost. This is achieved through triggering real-time interventions.

Accordingly, ongoing cases are continuously monitored during process operation, and decisions are made regarding whether to trigger an intervention. The intervention aims to reduce the cost associated with negative outcomes. However, it is important to recognize that while interventions may be beneficial, they also incur an associated cost, known as the cost of intervention. Therefore, the challenge is balancing the cost of the intervention with the cost of the negative outcome, ensuring that interventions are only triggered when needed, which is necessary for maximizing the total gain function.

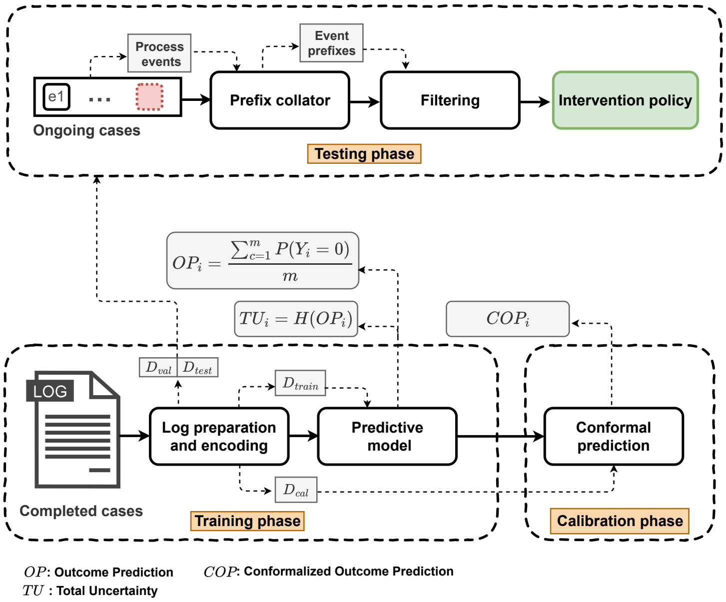


Figure 11. An overview of the uncertainty-aware prescriptive process monitoring approach (UN-PrPM).

The *UN-PrPM* develops an intervention policy that relies on predictive and uncertainty measures and consists of three: *training*, *calibration*, and *testing or operational*, as depicted in Fig. 11. Since the training and calibration phases are detailed in Sections 4.1 and 4.3, we will briefly describe their objectives to make the approach self-contained and focus more on the testing or operational phase.

4.4.1. Training and Calibration phases

The training and calibration phases aim to build an uncertainty-aware predictive model that provides estimates to determine which cases are suitable or in need of intervention. This is done by preparing and encoding event logs—and then training and tuning a predictive model. The trained predictive model aims to estimate the probability of a negative outcome for each case prefix, i.e., *outcome prediction (OP)*. Additionally, this model quantifies the *total uncertainty (TU)*, which arises either from the model's lack of knowledge or from the noise within the data itself. This model is then stacked with a conformal prediction method during the calibration phase to generate *conformalized outcome prediction (COP)*. The COP is a prediction set that contains the true outcome with a predefined confidence level. We decided not to implement the calibration step for the TU measure because TU already quantifies prediction uncertainty. However, the TU score could still give more information than the boolean information from the conformal prediction set. Thus, introducing an additional layer of uncertainty measurement, especially with the probability of a negative outcome, would enhance the intervention policy.

The combined *OP*, TU, and *COP* estimates are used to develop the intervention policy at the testing phase to maximize a total gain function while considering the cost associated with the intervention. By integrating predictive and uncertainty estimates, the *UN-PrPM* approach ensures that interventions are triggered only when necessary and with high confidence. However, it still does not consider the limited resources to execute the intervention addressed in the next chapters.

4.4.2. Testing Phase

In the testing phase, we operationalize the *UN-PrPM* approach. This phase leverages the estimates obtained from the training and calibration phases, i.e., *OP*,TU, and *COP*, to make real-time intervention decisions. During this phase, ongoing cases are continuously monitored to determine whether an intervention should be triggered based on the combined values of *OP*, TU, and *COP*. Here, we assume that cases arrive in a continuous stream, and we use a *prefix collator* to accumulate sequences of events. This process results in one or multiple ongoing cases at any given time, for which we must decide the necessity of triggering intervention.

To effectively manage this decision-making process, we propose an intervention policy formulated as a *filtering policy*. This filtering policy determines the suitability of applying an intervention to a given ongoing case. The filtering function, therefore, is a Boolean function that maps a case to either "suitable" or "not suitable" for intervention based on predefined *decision rules*. These decision rules, typically involving parameters and operators, are constructed based on the *OP*, TU, and *COP* estimates from the training and calibration phases. These decision rules filter ongoing cases into those eligible for interventions. In other words, the filtering policy eliminates cases that do not fulfill the decision rule.

A decision rule in a structured representation adheres to the following format: If a specific estimate satisfies a given condition (using an operator and a specified parameter threshold value), then the action is to include the case in the pool of suitable cases for intervention. For instance, a decision rule based solely on predictive estimates could be formulated as follows: If the outcome prediction (probability of a negative outcome) is greater than or equal to 0.5 (the threshold value), then the case is considered more likely to result in a negative outcome, and, accordingly, the intervention is triggered.

The *UN-PrPM* approach empowers business analysts with configurable intervention policies to maximize a total gain function. For instance, a business analyst aiming to trigger interventions for cases more likely to end in a negative outcome, with a specified confidence level of 80%, can set up the following decision rule: IF the outcome prediction (*OP*) is greater than 0.5 and the prediction set contains only the negative outcome with a significance level of $\alpha = 0.2$, then the case is considered suitable for intervention. This customizable approach enables analysts to align intervention policy criteria precisely with their organizational needs.

Accordingly, decision rules may have one or more parameters with corresponding threshold values. These threshold values are not determined beforehand and need to be optimized to achieve maximal total gain. One way to address this is to employ empirical thresholding, where different threshold values are tested, and the one that maximizes total gain is selected rather than relying on a random fixed value. This optimization ensures that cases are accurately targeted, increasing the number of cases ending with a positive outcome.

Once ongoing cases are defined as suitable for intervention, actions are triggered to transform cases likely to end in a negative outcome into a positive one. For example, in a loan origination process, a business case receives a loan offer and is estimated to be more likely to end with a negative outcome. Hence, another loan offer is prepared and sent to convince the case to accept it in a way that makes that case end with a positive outcome.

However, it is important to note that triggering interventions, while potentially beneficial and capable of achieving positive outcome gains, incur costs known as the cost of intervention. For instance, preparing and sending an additional loan offer involves administrative and processing costs in the loan origination process. These costs must be weighed against the potential gain from the positive outcome to ensure that the intervention is cost-effective.

4.5. Prescriptive Process Monitoring Evaluation

This section evaluates the *UN-PrPM* approach, which aims to develop an intervention policy (IP) that determines which cases need intervention and when to trigger it to maximize a performance measure, specifically total gain. The evaluation is conducted both with and without empirically optimizing the policies' parameter thresholds and by varying the ratio between the intervention costs and positive outcome gain. Thus, we further divide RQ1 into another three sub-research questions as follows:

RQ1.3: To what extent does empirical thresholding optimization impact the effectiveness of an intervention policy in terms of total gain?

RQ1.4: How does varying the ratio between intervention costs and positive outcome gain affect the total gain of an intervention policy?

RQ1.5: How does the success rate of interventions differ between policies that incorporate uncertainty measures and those that do not?

4.5.1. Datasets

We employed two publicly available datasets, namely *BPIC2012* and *BPIC2017*, which have been previously utilized in evaluating predictive monitoring approaches as discussed in Section 4.2.1. These datasets represent loan origination processes and have varying characteristics, as detailed in Table 7 and Fig. 7. They provide a clear notion of case outcomes and interventions that can positively impact these outcomes.

In this context, the outcome of each case is whether the loan application ends successfully or not, determined by a specific condition (boolean function) evaluated on a completed case. Outcomes are defined based on each case's last activity. If the last activity indicates a negative outcome, the intervention is considered to be sending another loan offer to the applicant.

The intervention is identified by the "Send_Offer" activity. This intervention aims to enhance the possibility of the applicant accepting a loan offer by making an additional offer. We count the number of offers sent to cases (or applicants), categorizing cases that receive only one offer into the control group (T=0) and cases that receive more than one offer into the treatment group (T=1). This categorization enables the evaluation of the impact of interventions on case outcomes.

4.5.2. Experimental setup

In our experimental setup, we used Python version 3.10. We employed a temporal split approach, where cases are sorted chronologically by their timestamps, to divide each log into three subsets: training (50%), validation (D_{val}: 25%), and testing (D_{test}: 25%). Additionally, we performed a further split within the training set to create a calibration set, resulting in the following splits: training set (D_{train}: 40%) and calibration set (D_{cal}: 10%).

The training set is used to train a predictive model, while the calibration set is utilized to obtain conformalized estimates with a guaranteed confidence level for outcome predictions. The validation and testing sets are used during the testing phase to operationalize the *UN-PrPM* approach and to learn the intervention policy. Each subset is composed of (X_i, T_i, Y_i), where X_i is a feature vector describing encoded inter-case and intra-case features by the index encoding technique, T_i is an intervention that positively impacts the outcome Y_i.

The predictive model is trained using a probabilistic ensemble learning technique [37]. This model estimates the total uncertainty (TU) in the predictions of a gradient-boosting classification model, specifically CatBoost [22], which provides the outcome prediction (*OP*), i.e., the probability of a negative outcome. To convert *OP* into a conformalized outcome prediction (*COP*), we follow the inductive conformal prediction approach using a Python package known as *ACPI*[4]. This package derives *COP*, which converts point estimates into prediction sets that contain the actual outcome with a predefined confidence level.

We operationalized the *UN-PrPM* approach to derive three intervention policies that differ based on the structure of decision rules used in the filtering step, as follows:

- **IP1 (Predictive only):** The decision rule relies solely on outcome prediction estimates. Ongoing cases are filtered into those needing intervention when the *OP* exceeds a certain threshold (τ_{OP}). The default τ_{OP} is set at 0.5, similar to binary classification tasks, but can vary between 0.1 and 0.9. This policy is a baseline referenced from the literature introduced in [46].
- **IP2 (Predictive plus conformal):** This policy extends IP1 by incorporating both the OP and the COP to filter ongoing cases. Interventions are triggered when $OP > 0.5$ and the $COP = 1$, which indicates that the prediction set contains only the negative outcome, i.e., $COP = \{0\}$. The confidence level (τ_{COP}) influences the constructed prediction set, with a default value set at 0.9, indicating high confidence. However, τ_{COP} can vary from 0.1 to 0.9. Lower confidence levels indicate higher α and vice versa.
- **IP3 (Predictive plus conformal plus total uncertainty):** This policy integrates *OP*, *COP*, and TU measures to provide a stricter intervention policy. Cases are filtered based on three conditions: $OP > 0.5$, $COP = 1$, and TU is less than an uncertainty threshold (τ_{TU}). The default τ_{TU} value is set at 0.5, indicating an average level of uncertainty, and can vary between 0 (low uncertainty) and 1 (high uncertainty).

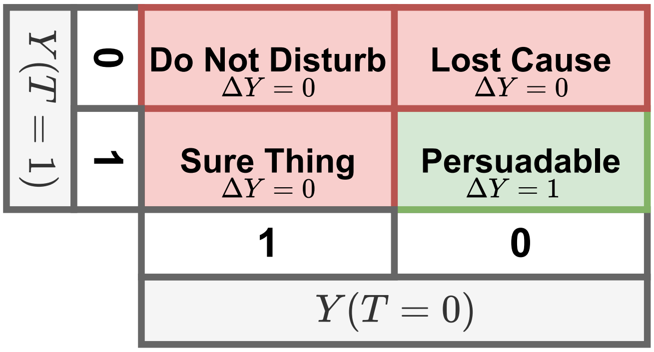
In our experiments, we use the validation set (D_{val}) to empirically optimize the corresponding thresholds for each policy to maximize a total gain function. For IP1, we vary the τ_{OP} value between 0.3 and 0.9 to distinguish between relaxed and strict policies. For IP2, we optimize both τ_{OP} and τ_{COP} to find the best confidence level and outcome prediction values that yield the maximum total gain on the validation set. Similarly, for IP3, in addition to optimizing τ_{OP} and τ_{COP}, we optimize τ_{TU}, setting the range between 0.1 and 0.9. We use the TPE algorithm, conducting 50 trials to find the optimal threshold values for each policy.

Performance Measures. An intervention policy, e.g., IP1, IP2, and IP3, aims to target cases where, without intervention, the outcome is negative, but with intervention, the outcome becomes positive. We refer to such cases as persuadable cases, as illustrated in Fig. 12. In other words, these are cases where $Y(T = 0) = 0$

---

[4] `https://github.com/salimamoukou/ACPI`

and $Y(T = 1) = 1$. Hence, it is necessary to avoid other situations where cases belong to the "do not disturb," "sure thing," or "lost causes" groups.



		$Y(T=0)$	
		1	0
$Y(T=1)$	0	**Do Not Disturb** $\Delta Y = 0$	**Lost Cause** $\Delta Y = 0$
	1	**Sure Thing** $\Delta Y = 0$	**Persuadable** $\Delta Y = 1$

Figure 12. Grouping of cases based on intervention outcomes

We incorporate this change in outcome (ΔY) into the goodness measure of intervention policies, i.e., the *total gain*. The ΔY can have two possible values, zero or one. It takes the value of one when the outcome without intervention is negative and becomes positive with the intervention; otherwise, it takes the value of zero. Additionally, the total gain function considers two other parameters: the gain of having a positive outcome (*Gain*) and the intervention cost (C_{in}), since triggering interventions could be beneficial but at the same time incurs a cost, as shown in Eq. 4.4. Hence, to explore the impact of different intervention costs, we experiment with varying the ratio of intervention cost to positive outcome gain, such as 100%, 90%, down to 10%.

$$TotalGain = (\Delta Y * Gain) - C_{in} \tag{4.4}$$

$$\Delta Y = \begin{cases} 1 & \text{if } Y(T=0) = 0 \text{ and } Y(T=1) = 1 \\ 0 & \text{otherwise} \end{cases}$$

The total gain function assumes knowledge of both outcomes (with and without intervention) for each case. However, real-life event logs provide information on only one of the two possible outcomes per case: (1) If an intervention occurred, the outcome given the intervention is known, but the outcome without the intervention is unknown; and (2) If an intervention did not occur, the outcome given no intervention is known, but the outcome with intervention is unknown. Inspired by [1], we employ an alternative outcome estimator called *RealCause* [30] to address this information gap. This estimator is utilized when only one possible outcome is known per sample (i.e., either the outcome given an intervention occurred or the outcome given no intervention). RealCause estimates the alternative outcome by assuming that case outcomes follow a binary or Bernoulli distribution.

4.5.3. Results

This section compares intervention policies developed using the *UN-PrPM* approach, specifically IP1, IP2, and IP3. The evaluation focuses on the total gain function while varying the ratio of intervention cost to positive outcome gain. We also assess the impact of empirical thresholding on parameter optimization for each policy's effectiveness.

Figures 13 and 14 illustrate the performance of different policies under various cost-to-gain ratios, addressing RQ1.4, and the effect of empirical thresholding, addressing RQ1.3, on total gain for the BPIC2012 and BPIC2017 event logs, respectively. The results are categorized into three distinct groups based on the ratio of intervention cost to positive outcome gain:

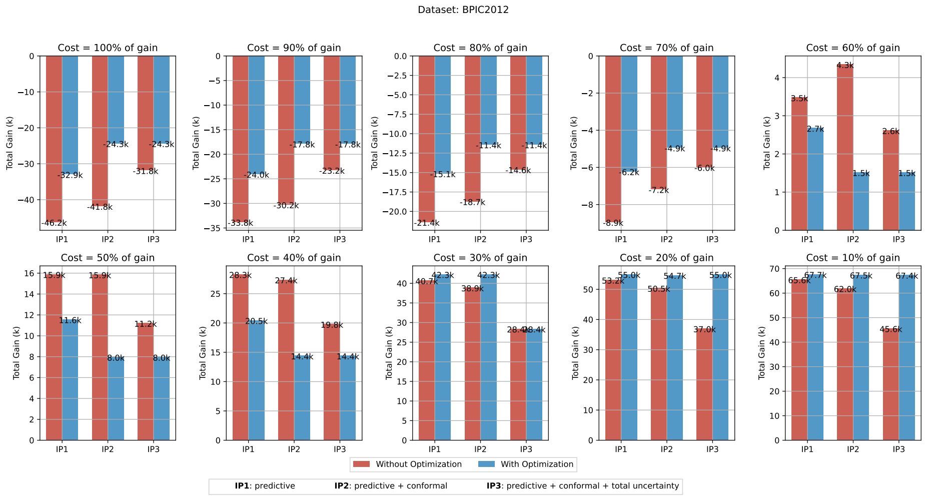


Figure 13. Effect of empirical thresholding and Performance of different policies under various cost-to-gain ratios for the BPIC2012 event log.

- **High Intervention Cost Relative to Positive Outcome Gain ($C_{in} \geq 70\%$):** In this scenario, none of the policies (IP1, IP2, or IP3) achieve a positive total gain, regardless of whether empirical thresholding is applied. This indicates that achieving a positive total gain in settings with high intervention costs is challenging and requires highly accurate intervention decisions. Policies incorporating conformal and uncertainty estimates (IP2 and IP3) tend to incur fewer negative losses or result in zero gain than pure predictive policy (IP1). This suggests that integrating uncertainty measures makes intervention policies more conservative, reducing negative losses or zero gain in the best case. However, empirical thresholding helps to decrease negative losses, demonstrating an improvement over policies without empirical thresholding.

- **Moderate Intervention Cost Relative to Positive Outcome Gain** ($40\% \geq C_{in} \leq 60\%$)**:** Here, the intervention cost is roughly half of the positive outcome gain, creating a balanced ratio. As a result, the benefits of correctly targeting interventions are comparable to the costs of making incorrect decisions. This situation is somewhat analogous to random guessing, where achieving high performance depends on the accuracy of the guesses. For instance, IP1 performs better under these conditions as it does not incorporate additional constraints. Conversely, when conformal and uncertainty measures are integrated (as in IP2 and IP3), the intervention decisions are restricted, potentially leading to missed opportunities for achieving higher total gains. Moreover, policies that do not utilize empirical thresholding generally outperform those that incorporate empirical thresholding in this balanced cost-to-gain scenario.

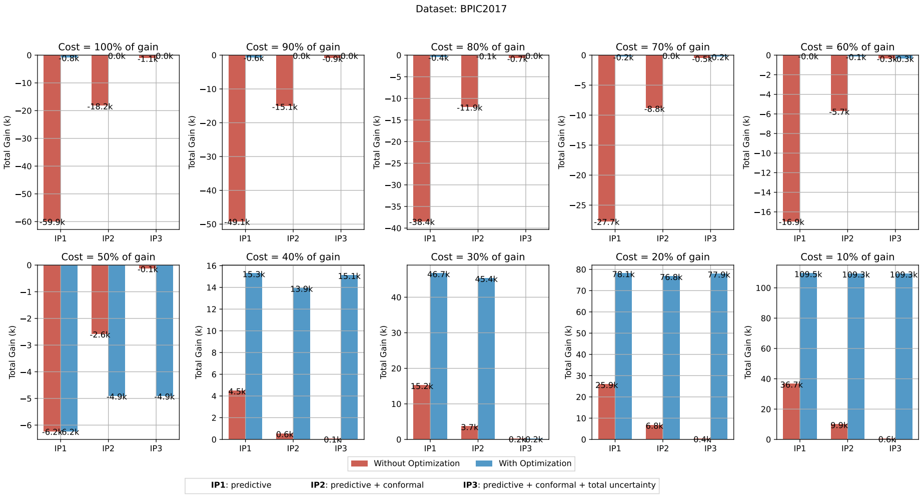


Figure 14. Effect of empirical thresholding and Performance of different policies under various cost-to-gain ratios for the BPIC2017 event log.

- **Low Intervention Cost Relative to Positive Outcome Gain** ($C_{in} \leq 30\%$)**:** In this case, the intervention cost is relatively low compared to the positive outcome gain. As a result, policies incorporating conformal and uncertainty estimates (IP2 and IP3) achieve performance comparable to that of purely predictive estimates (IP1) when empirical thresholding is applied. However, without empirical thresholding, the purely predictive policy (IP1) outperforms the other policies. This indicates that when the intervention cost is low relative to the potential gain, the benefits of a purely predictive approach become more apparent, particularly when empirical thresholding is not used. However, empirical thresholding significantly enhances performance in low-cost scenarios, and a straightforward predictive approach can be more effective.

The above analysis shows that policies relying on conformal and uncertainty estimates become more conservative, especially when the intervention cost is relatively high compared to the potential outcome gain. This observation leads us to analyze the success rate of triggering interventions to address RQ1.5 and to understand the efficiency and precision of different intervention policies, particularly those incorporating conformal and uncertainty estimates.

The success rate is the proportion of triggered interventions that result in a positive outcome. Specifically, the success rate measures the effectiveness of interventions in converting cases that would otherwise end negatively into positive outcomes. In other words, it quantifies the proportion of persuadable cases that change from a negative to a positive outcome due to the intervention.

Figures 15 and 16 illustrate the number of interventions triggered under each policy, both with and without empirical thresholding, for the BPIC2012 and BPIC2017 event logs, respectively. Table 10 presents the success rate for each policy, comparing the results with and without empirical thresholding and across varying cost-to-gain ratios.

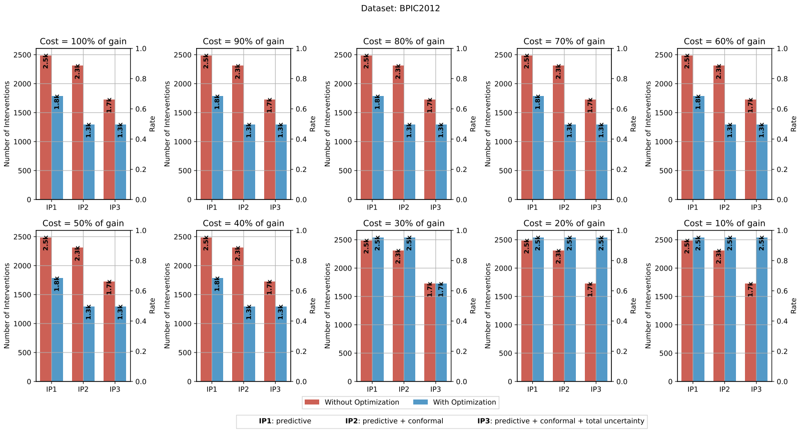


Figure 15. Effect of empirical thresholding on the number of triggered interventions of different policies under various cost-to-gain ratios for the BPIC2012 event log.

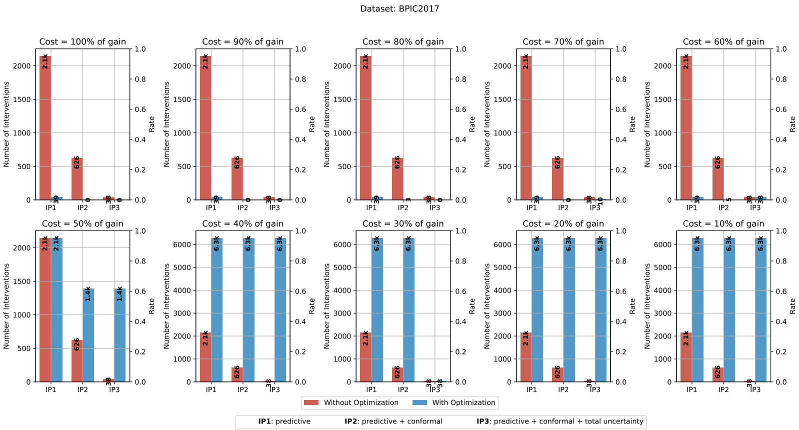


Figure 16. Effect of empirical thresholding on the number of triggered interventions of different policies under various cost-to-gain ratios for the BPIC2017 event log.

This analysis reveals that policies incorporating conformal and uncertainty estimates (IP2 and IP3) trigger fewer interventions than those relying solely on predictive estimates (IP1), especially when the intervention cost is relatively high or moderate compared to the positive outcome gain. However, when the intervention cost is low, all policies trigger similar interventions, indicating a more relaxed approach.

Empirical thresholding significantly impacts the number of triggered interventions. When the intervention cost is high or moderately high, empirical thresholding constrains the policies, resulting in fewer triggered interventions. Conversely, when the intervention cost is low, empirical thresholding allows for more interventions.

Additionally, empirical thresholding enhances the success rates of all policies compared to when it is not applied. Notably, policies with uncertainty measures (IP2 and IP3) outperform those without in terms of success rates, as shown in Table 10. Hence, integrating uncertainty measures with predictive measures results in fewer interventions but with higher success rates.

Table 10. Success rates of different policies under various cost-to-gain ratios with and without empirical thresholding

Dataset	Optimization	Policy	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
BPIC2012	With	IP1	0.6298	0.6298	0.6298	0.6298	0.6298	0.6298	0.6298	0.6333	0.6333	0.6333
		IP2	0.6431	0.6431	0.6431	0.6431	0.6431	0.6431	0.6431	0.6533	0.6510	0.6517
		IP3	0.6431	0.6431	0.6431	0.6431	0.6431	0.6431	0.6431	0.6501	0.6533	0.6509
	Without	IP1	0.6278	0.6278	0.6278	0.6278	0.6278	0.6278	0.6278	0.6278	0.6278	0.6278
		IP2	0.6375	0.6375	0.6375	0.6375	0.6375	0.6375	0.6375	0.6333	0.6309	0.6317
		IP3	0.6301	0.6301	0.6301	0.6301	0.6301	0.6301	0.6301	0.6301	0.6333	0.6309
BPIC2017	With	IP1	0.3641	0.3641	0.3641	0.3641	0.3641	0.2418	0.2486	0.2486	0.2486	0.2486
		IP2	-	-	0.4000	-	0.4193	0.4293	0.4443	0.4443	0.4443	0.4481
		IP3	-	-	-	0.4000	0.4211	0.4293	0.4481	0.4211	0.4481	0.4481
	Without	IP1	0.2418	0.2418	0.2418	0.2418	0.2418	0.2418	0.2418	0.2418	0.2418	0.2418
		IP2	0.3169	0.3169	0.3169	0.3169	0.3169	0.3169	0.3169	0.3169	0.3169	0.3169
		IP3	0.3211	0.3211	0.3211	0.3211	0.3211	0.3211	0.3211	0.3211	0.3211	0.3211

4.5.4. Threats to Validity

The evaluation of the *UN-PrPM* approach poses two main threats to external validity, as follows:

- **Limited Event Logs Utilization.** The use of only two datasets limits the generalizability of our evaluation. This constraint is due to the experimental setup's requirement for datasets that contain both case outcomes and interventions, ensuring a causal relationship between the intervention and the outcome. Despite this limitation, the two selected event logs (BPIC2012 and BPIC2017) exhibit different characteristics, providing some diversity in the evaluation. Additionally, we checked the datasets in the 4TU Centre for Research Data[5] and in previous studies on prescriptive process monitoring but did not find other event logs fulfilling these requirements. We opted not to inject synthetic interventions, as this would not have demonstrated the potential benefits of the approach in real applications.
- **Method of Estimating Alternative Outcomes.** Another threat to external validity arises from the method used to estimate alternative outcomes, specifically RealCause [30], which extrapolates the outcome if an intervention had not occurred. While RealCause has a well-established theoretical foundation and has undergone extensive evaluation, it may not accurately estimate alternative outcomes in all cases, introducing a potential source of error. This inaccuracy could impact the validity of the intervention policy's effectiveness, particularly in diverse or unseen contexts.
- **Concept Drift** Concept drift in business processes occurs when process dynamics evolve over time due to changing business needs, external influences, or shifts in user behavior [318]. This creates a challenge for PrPM, as intervention policies based on historical data may become less effective when encountering new, previously unseen patterns. If not properly addressed, concept drift can lead to inaccurate or outdated prescriptive recommendations, compromising decision-making reliability and effectiveness. To mitigate this risk, adaptive mechanisms such as continuous model updates, periodic retraining, incremental learning, and drift detection techniques are essential to maintain the relevance of intervention strategies [319]. However, addressing concept drift is beyond the scope of this PhD thesis and remains an avenue for future research.

---

[5] `https://data.4tu.nl/datasets/5ea5bb88-feaa-4e6f-a743-6460a755e05b`

4.6. Summary

This chapter presents an uncertainty-aware prescriptive process monitoring (UN-PrPM) approach to enhance business process performance by triggering run-time interventions. The *UN-PrPM* approach aims to develop an intervention policy capable of confidently identifying cases likely to end negatively and then triggering interventions to transform these outcomes into positive ones. To achieve this, predictive monitoring is integrated with uncertainty measures, facilitating the creation of uncertainty-aware intervention policies.

On the one hand, this chapter examines different predictive process monitoring (PPM) approaches and methods for measuring uncertainty in predictions. It benchmarks various PPM approaches to identify the most accurate predictive model for use in prescriptive settings. The selected model is further enhanced with two methods to measure uncertainty, ensuring high-quality predictions and subsequent intervention decisions.

On the other hand, the chapter introduces the uncertainty-aware prescriptive process monitoring approach to develop an intervention policy to maximize a total gain function. This policy assumes no resource constraints, similar to the existing literature, allowing for immediate and unlimited interventions. However, it differs from existing literature by being uncertainty-aware.

We evaluated the *UN-PrPM* approach on real-life event logs. The evaluation confirms that uncertainty-aware policies are generally more conservative, resulting in a comparable total gain with fewer interventions but higher success rates. Accordingly, this approach could be suitable for environments with limited resources where interventions cannot be triggered immediately for all cases or in large numbers.

In the next chapter, we will explore the development of intervention policies in a constrained resource environment. We will extend the *UN-PrPM* approach to account for resource limitations and adjust it accordingly.

5. PRESCRIPTIVE PROCESS MONITORING UNDER RESOURCE CONSTRAINTS: A WHITE-BOX APPROACH

In the previous chapter, we introduced an uncertainty-aware prescriptive process monitoring (UN-PrPM) approach that identifies, with guaranteed confidence, cases likely to end negatively during business process execution and triggers interventions to transform these outcomes into positive ones, thereby maximizing a total gain function. However, as common in existing PrPM literature, this approach assumes the ability to trigger interventions immediately and for all identified cases, implying unlimited resource availability. In practice, interventions require time and commitment from scarce resources, such as a loan officer's time to prepare a second loan offer in a loan origination process. When a PrPM approach triggers an intervention, it occupies a resource that is unavailable for other potentially higher-gain interventions for a certain duration. Therefore, in a resource-limited environment, a PrPM approach must consider not only the necessity, timeliness, and effect of interventions but also the uncertainty of these measures and the level of resource utilization. Committing scarce resources to uncertain intervention decisions can lead to suboptimal policies.

Learning an intervention policy from past observations is the backbone of a PrPM approach. Such a policy determines who requires intervention, when the intervention should be triggered, and who should execute it at a given state of a process case. Accordingly, an intervention policy needs to consider parameters such as the probability of a case ending positively or negatively, the time remaining to trigger an intervention, and the expected effectiveness of the intervention in changing the case outcome. Additionally, the decision to trigger interventions must consider the availability of the human resources needed to perform the intervention and the level of uncertainty to ensure efficient resource allocation.

Furthermore, in some business scenarios, regulatory requirements mandate that automated decisions affecting individuals should not be based on profiling, as stated in Article 22 of the General Data Protection Regulation – GDPR) [183]. Consequently, business stakeholders must be able to explain the basis for intervention decisions made by a prescriptive monitoring system and adjust these decisions' parameters to prevent unwanted biases. This necessitates a white-box approach to defining intervention policies for PrPM.

In this setting, this chapter addresses a specific instance of the general PrPM problem: *Given past observations of completed cases, where some cases had interventions, and each intervention has a cost and consumes resources with limited capacity—the research question is posed as follows:* ***RQ2:*** *How can business stakeholders define transparent, rule-based white-box intervention policies that manage limited resources efficiently while optimizing the total gain of the resulting intervention policy in a resource-constrained setting?*

To answer this question, we present *WB-PrPM* (White-Box Prescriptive Process Monitoring), a rule-based framework that enables stakeholders to create transparent and configurable intervention policies. The white-box nature stems from the transparency in how the intervention policy is developed during process execution rather than from the underlying techniques themselves. WB-PrPM integrates various machine learning techniques to estimate key factors influencing intervention decisions, such as *need*, *effect* (importance of the intervention), and time left to intervene or *intervention window* (urgency of the intervention). The framework also incorporates uncertainty associated with these factors (need, effect, and intervention window) to ensure confident intervention decisions. Additionally, WB-PrPM includes an automated parameter optimization method and uses filtering and ranking mechanisms alongside a resource allocator to prioritize cases, ensuring that limited resources are allocated effectively to maximize overall gain. This extended approach builds upon existing research in PrPM [46, 61, 59], focusing on the importance, uncertainty, urgency, and capacity dimensions to execute interventions.

We report on an empirical evaluation to assess the effectiveness of different framework variants, particularly in terms of total gain (i.e., the sum of the benefits of the interventions triggered by the policy minus the sum of the intervention costs) across various resource utilization levels. This evaluation compares white-box-based baseline methods regarding the intervention policy but lacks consideration for resource constraints and inherent uncertainty parameters within the policy [46, 61, 59]. The objective is to explain how intervention policies under limited resources perform in terms of total gain.

We begin the chapter by introducing the core concepts used in our approach in Sections 5.1 and 5.2 and then move on to present the approach in Sections 5.3 and 5.4, which is followed by the empirical evaluation in Section 5.5. Finally, the chapter is summarized in Section 5.6.

5.1. High-Level Approach Overview

The *WB-PrPM* approach is designed to handle sequential decision-making in resource-constrained environments with uncertainty. Its main goal is to develop an intervention policy that effectively balances the need for interventions with the limitations of available resources. At a high level, this approach is structured around three key components: *identifying factors influencing intervention policy decisions*, *an offline phase*, and *an online phase*.

The first key component is to *identify factors that impact intervention policy decisions*. Some of these factors are mainly driven by previous work in PrPM [202, 285, 251, 317, 315]. The rest are imported from the literature on decision support systems and operational research to cope with scenarios where resources are limited [306, 303, 300]. These factors may sway the policy decision to trigger an intervention and allocate resources to a specific case within a given state. These factors are depicted in Fig. 17, and we categorize them into four top-level dimensions relevant to the intervention policy: *Importance*, *Urgency*, *Capacity*, and *Uncertainty*. For instance, within a loan origination process, such factors may include the need for and effect of sending an additional loan offer to applicants and the allowable time frame for executing the intervention. Identifying such factors is crucial for determining cases that require interventions and helps in case prioritization, thereby enabling the efficient allocation of limited resources.

The second key component is an *offline phase* (see Fig. 18), wherein various machine learning models are trained to parametrize these factors and estimate them during process execution. Specifically, a *predictive model* is employed to estimate the need for intervention, while a *causal model* is utilized to estimate the effectiveness of the intervention. Both factors, need and effect, denote the importance dimension of the intervention. Additionally, a *survival model* is trained to estimate the intervention window factor, i.e., the remaining timeframe for intervention execution to be feasible, representing the urgency dimension. These estimates are then calibrated to provide conformalized estimates with more confidence than low-quality point estimates, ensuring efficient decision-making during the online phase.

We instantiate and operationalize the approach in the third component, the *online phase* (as shown in Fig. 18). Here, the trained models estimate intervention policy decisions' parameters. These estimates, combined with a *resource monitoring and allocation* component representing the capacity dimension, simulate the constrained resources' environment, thus enriching the learning process for an intervention policy. Also, the factor parameters are empirically optimized to find the best parameter threshold values, which are then used to evaluate the intervention policy in terms of a performance measure, i.e., total gain.

5.2. Factors Affecting Intervention Policies

The WB-PrPM is built around key factors and designed to address decision-making in a resource-constrained environment. While some of these factors are adapted from existing literature [202, 285, 251, 317], they have been extended to better suit situations with limited resources [306, 303, 300]. These factors guide decisions on triggering interventions and resource allocation for specific cases at a given state. As shown in Fig. 17, the factors are categorized into four main dimensions: *Importance* (including *need* and *effect* factors), *Urgency* (represented by the *intervention window* factor), *Capacity* (reflecting resource availability), and *Uncertainty* (represented by conformal estimates).

5.2.1. Importance

The dimension of importance addresses the question of whether an intervention is both needed and effective for a particular case. Interventions should be triggered when necessary, indicating that cases will likely end negatively. Additionally,

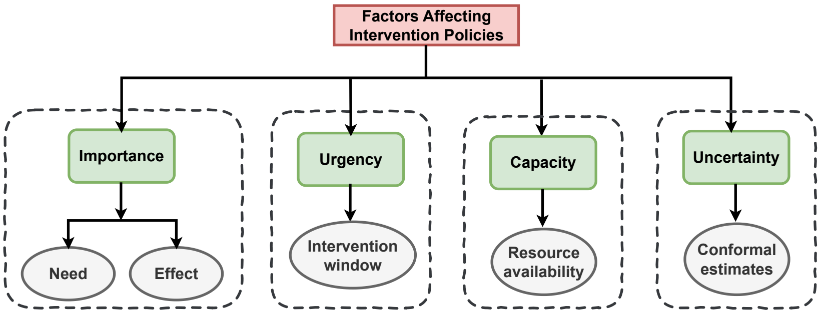


Figure 17. Factors influencing intervention policy decision-making.

these interventions have the potential for significant effect, capable of preventing negative outcomes and thereby delivering value.

The *need* for intervention is determined by reaching a specific confidence level in predicting a negative outcome for a case [202], ensuring high confidence in intervention necessity. We use a dual-factor method to quantify this need factor. The first parameter estimates the probability of cases likely to end with a negative outcome, known as outcome prediction (OP) [270]. The second parameter is the total uncertainty (TU) inherent in these predictions [315], acknowledging that predictions are not always perfect and may have varying degrees of uncertainty. Including total uncertainty helps prioritize some cases over others, regardless of their outcome prediction. An *outcome-oriented predictive model* facilitates this assessment by providing a reliable estimate of the probability of a negative outcome (OP) and the total uncertainty (TU) associated with the prediction.

The *effect* of an intervention (IE) addresses how significantly the probability of a positive case outcome increases upon triggering the intervention [285, 317]. This directly relates to the effectiveness of the intervention. We use a *causal model* to estimate the intervention's effect, providing a parameter for this factor. This evaluation determines whether triggering the intervention has the potential to transform a case with a negative outcome into a positive one. Providing the policy with information regarding the expected effect of the intervention helps develop a more optimal intervention policy.

For example, in a loan origination process, assume an applicant has a 55% probability of a positive outcome and a 45% probability of a negative outcome, with 70% total uncertainty. If we estimate the probabilities of the outcome with and without intervention as 45% and 80%, respectively, the resulting impact of triggering the intervention would be an increase in the probability of a positive outcome by 35% and a decrease in the probability of a negative outcome by 35%. Hence, the effect factor highlights the intervention's ability to transform a negative case outcome into a positive one.

5.2.2. Urgency

The urgency dimension focuses on determining the optimal timing for interventions or prioritizing cases that require immediate action. This concept is borrowed from decision support systems, where it is used, for instance, to assist managerial decision-making in fields like healthcare and targeted interventions [306]. In our context, urgency helps decide whether to intervene immediately, delay intervention to a later stage, or not intervene at all. This dimension is quantified by the *Intervention Window* (*IW*), which represents the remaining time before an intervention becomes unfeasible. For example, in a loan application process, the *IW* begins when the application is submitted and lasts until the estimated time of a potentially negative outcome, such as the rejection of a loan application within five days.

In a resource-constrained environment, consider a scenario where only one resource is available for interventions, such as a single loan officer handling urgent cases. Suppose three cases require immediate intervention due to their time-sensitive nature. Cases one and two are more likely to end with a negative outcome and slightly more significant intervention effects than case three. However, case three is expected to end negatively much sooner than the other two, making it more urgent. An optimal policy would prioritize case three due to its temporal urgency and postpone interventions for cases one and two. Therefore, an ML model that can predict the close negative outcome in case three while suggesting postponing for cases one and two is necessary.

A *survival model*, which specializes in estimating the duration until a specific event, as a negative outcome occurs, offers a fitting approach for estimating the intervention window. This model resembles a PPM task, specifically a remaining time prediction. However, it differs from the PPM task in that it focuses on estimating the time until a negative event occurs rather than simply predicting the remaining time without considering whether the last event is negative or positive. Consequently, and for prescriptive settings, it is important to focus on the time left before a negative outcome occurs, providing a more related estimate and, thus, an accurate intervention decision.

5.2.3. Capacity

The capacity dimension focuses on the feasibility of executing interventions based on available resources [303]. Even when an intervention is necessary, effective, and urgent, it may be impossible to implement if there aren't enough resources. Therefore, monitoring and managing resource availability is required to ensure the successful execution of interventions.

For example, a loan officer might need to contact applicants to offer an alternative loan option in a loan origination process. If there is only one officer available and multiple cases need immediate attention, the officer can handle only one case at a time. To model this, we introduce a factor (n) representing the number of available resources. During the operationalization of the WB-PrPM, resources are continuously monitored. When a case requires intervention and a resource is available, it is allocated to that case. The resource remains occupied for the duration of the intervention and becomes available for the next case once completed. This approach ensures that interventions are feasible and timely, given resource constraints.

5.2.4. Uncertainty

The uncertainty dimension in PrPM addresses the challenge of developing intervention policies based on low-quality estimates or that inherently come with uncertainty, indicating less confidence in the guarantees provided by these estimates [300]. In the context of resource constraints, the key dilemma in PrPM is deciding when to trigger interventions, considering not only concerns regarding the need, effect, or time left to intervene, but also the uncertainty associated with these estimates.

Committing limited resources to an intervention becomes problematic when the need or effect of the intervention is highly uncertain. For instance, when estimating the need and effect of an intervention, predictive and causal models assess the probability of a negative outcome and the expected intervention effect. However, these point estimates are often inherently uncertain, potentially leading to unnecessary interventions and wasted efforts. In resource-limited settings, relying on uncertain estimations of factors for resource allocation can be particularly problematic, potentially leading to inefficiencies and suboptimal policies.

To address these uncertainties, we employ *conformal prediction*, a model-agnostic framework that transforms point estimates into *prediction sets* (for classification tasks) or *prediction intervals* (for regression tasks). This method ensures that the true outcome is included within the prediction interval or set with a predefined confidence level $(1-\alpha)$, thereby increasing the reliability of intervention decisions. Unlike other uncertainty quantification methods, such as those proposed in [203], conformal prediction does not quantify the level of uncertainty itself but provides a guarantee that predictions meet a specified confidence threshold [268]. For example, conformal prediction can be used to assess the likelihood of a loan rejection with a 95% confidence interval in a loan origination process.

After detailing the dimensions of importance (need and effect), urgency (intervention window), capacity (resource availability), and uncertainty (conformal measures), along with their corresponding factors influencing intervention decisions, we now proceed to discuss the offline phase and the online phase of the WB-PrPM approach, as illustrated in Fig. 18.

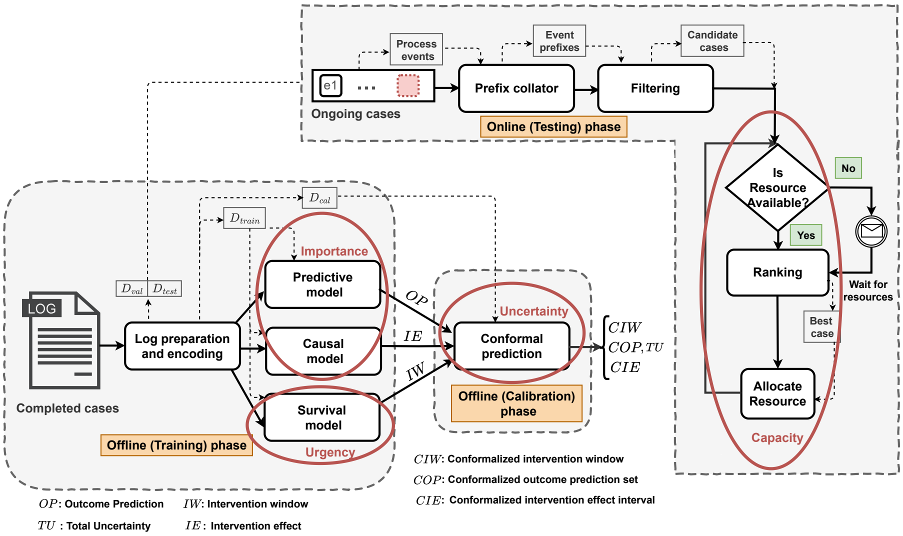


Figure 18. An overview of the white-box prescriptive process monitoring (WB-PrPM) approach.

5.3. Offline phase of the WB-PrPM

The offline phase of the *WB-PrPM* approach is essential for setting the foundation of the intervention policy by utilizing historical event logs, representing historical business process execution, and training machine learning models for decision-making. This phase is divided into two sub-phases: *Training* and *Calibration*.

In the Training sub-phase, event logs are prepared and encoded, and several models are trained, including predictive, causal, and survival models. These models capture intervention policy factors, such as predicting need, estimating intervention effects, and assessing the urgency of interventions. While the log preparation and encoding and the training of a predictive model were discussed in detail in the previous chapters, specifically in Section 4.1.1, we will briefly revisit them before focusing on the other components.

The Calibration sub-phase ensures that the predictions from these models are calibrated to account for uncertainty by generating conformalized estimates instead of point estimates, as discussed in Section 4.3.2. These conformalized estimates will later be applied in the online phase to guide white-box intervention policies.

5.3.1. Log preparation and encoding and predictive model.

The first step in the training phase involves preparing and encoding event logs to make them suitable for training ML algorithms. This process encompasses data cleaning, feature engineering, and sequence encoding, as detailed in Section 4.1.1. The outcome of this step is an encoded prefix log, where each prefix belongs to a specific business case and includes a fixed-size set of inter-case and intra-case features, along with a case outcome and an intervention that could potentially convert negative outcomes to positive ones. This encoded prefix log is then divided into separate subsets to be utilized in the different phases of the prescriptive approach, as shown in Fig. 18. For instance, a training set is used to fine-tune and train the ML models.

Among these models is the *predictive model*, which estimates the probability of cases ending in a negative outcome at various prefixes (or states) within a given case. This is conducted by training an ensemble learning method, specifically the gradient boosting classification algorithm, CatBoost, with two primary objectives. The first objective is to predict the probability of a negative outcome for each case prefix, referred to as outcome prediction (OP, as shown in Eq. 4.1), as accurately as possible. The second objective is to quantify the total uncertainty (TU, as in Eq. 4.2) associated with these predictions, which may stem from the model's limited knowledge or inherent noise within the data, as shown in Section 4.3. The OP and the TU parameters quantify the *need* for intervention in prescriptive settings.

Still, different methods can be used to estimate the probability of a negative case outcome. The specific method used for this purpose is part of the instantiation of the WB-PrPM approach. We opt for an ensemble method (gradient boosting) in the empirical evaluation reported later in this chapter. However, in use cases where there is a strong white-box requirement, a white-box prediction method could be used instead (e.g., decision trees) with the usual tradeoffs between accuracy and interpretability.

5.3.2. Causal Model

The causal model's objective in the training phase is to estimate the causal relationship between an intervention or treatment (T) and business case outcomes. The focus is on evaluating whether the intervention effect (IE) can transform a business case outcome from negative to positive. This effect is determined by comparing the probabilities of the outcome occurring with the intervention ($Y(T=1)$) against the probabilities without the intervention ($Y(T=0)$). For instance, if the probability of accepting a loan offer is 0.9 with the intervention and 0.3 without, the IE is quantified as 0.6. This estimation helps understand whether triggering an intervention will reduce the occurrence of negative outcomes.

We use the potential outcome framework to estimate the intervention effect, which can be expressed through various measures, including individual treatment effects (ITE, as in Eq. 2.1) and conditional average treatment effects ($CATE$, as in Eq. 2.2). While both ITE and $CATE$ are used in the literature to evaluate the impact of interventions, we focus on ITE due to certain limitations associated with $CATE$, as discussed in [184].

The $CATE$ does not account for individual variations in response to interventions, affecting intervention decisions. Additionally, accurate estimation of $CATE$ frequently requires access to extensive and sometimes difficult-to-obtain data. In contrast, ITE is better suited for scenarios with limited data availability and individual response variability. For example, ITE allows for assessing how a marketing campaign impacts each customer, capturing variations that $CATE$ might miss. Therefore, with its adaptability to practical data constraints and individualized responses, ITE could be more effective in developing intervention policies.

Again, the choice of the specific model to estimate the IE is part of the instantiation of the WB-PrPM approach. In the empirical evaluation, we use orthogonal random forest (ORF) as it handles high-dimensional feature spaces well. This is useful in our setting because event logs have many attributes with categorical values, leading to feature explosion. However, in the presence of a strict white-box requirement, one could use uplift trees instead, as proposed in [85].

5.3.3. Survival model

The survival model estimates the time until a specific event, such as a negative outcome, occurs [283]. It is useful for binary events, helping to understand time-to-event occurrence. The model parameterizes the *intervention window* (IW) in an intervention policy. In the loan origination process, for example, it estimates the risk function, representing the instantaneous rate at which a loan application can be rejected or declined at any given time, given that the applicant has not yet experienced the negative event.

Survival analysis is especially useful in resource-limited contexts, as it helps prioritize cases for timely interventions, thus preventing costly negative outcomes [306]. It balances the costs associated with intervening too early or too late, emphasizing the importance of timing for effective resource allocation. Therefore, the survival model determines an intervention window (IW). The IW for a given case i at a particular prefix is defined as the time frame within which an intervention can be executed before the negative outcome occurs, as described in Eq. 5.1. Here, the IW begins at 0, i.e., the current time, because interventions can be implemented immediately, and it ends at the window time (WT), representing the estimated time until the negative outcome is expected to occur for the case i.

$$IW_i = [0, WT_i] \tag{5.1}$$

For example, consider two loan origination cases, $C02$ and $C03$. If the current time for case $C02$ is 5 days and the negative outcome is predicted to occur at $WT = 10$ days, the IW spans from 5 to 10 days. This indicates a 5day window for implementing an intervention to mitigate the negative outcome. In contrast, if the current time for case $C03$ is 7 days and the negative event is predicted to occur at $WT = 10$ days, the IW spans from 7 to 10 days, signaling a more urgent need for intervention.

This approach enables the prioritization of cases based on urgency, ensuring that those with imminent negative outcomes receive attention first. Estimating the time until a loan decision (e.g., rejection) helps optimize resource allocation within lending institutions, ensuring that interventions are timely.

Unlike traditional remaining time predictions in PPM, which estimate time to case completion regardless of the outcome, survival analysis focuses specifically on predicting the time until a negative event occurs. Survival analysis also accounts for censoring, where some cases may not experience the event by the end of the process, adding complexity to the modeling process.

Common methods in survival analysis include *Kaplan-Meier estimation* [33],*Cox proportional hazards [34]*, or *parametric survival models* [35]. In contrast, traditional PPM often relies on regression-based techniques or machine learning algorithms to predict continuous time durations. In this study, we use the *Cox proportional hazards* model for its flexibility, as it does not require assuming a specific survival distribution [36]. This model utilizes a hazard function to assess how case characteristics relate to the time until an event occurs, offering a nuanced approach to predicting time-to-event data.

5.3.4. Calibration: Conformal prediction.

In the calibration phase, we apply conformal prediction to produce estimates with guaranteed confidence levels [130, 120, 121], as discussed in Section 4.3.2. This technique is applied on top of predictive, survival, and causal estimates—specifically, Outcome Prediction (OP), Intervention Window (IW), and Intervention Effect (IE)—to ensure that each estimate meets a user-defined confidence level, $1-\alpha$. Conformal prediction transforms these estimates into valid prediction sets or intervals. For example, OP becomes Conformalized Outcome Prediction (COP), IW becomes Conformalized Intervention Window (CIW), and IE becomes Conformalized Intervention Effect (CIE). Conformal prediction provides finite-sample guarantees that the prediction set or interval will contain the true outcome with the desired confidence level, i.e., $1-\alpha$, compared to heuristic uncertainty quantification methods [119, 313].

After calibration, the estimates are prefixed with "C" to signify their conformalized status. This approach replaces single-point estimates, which may have low confidence, with more reliable conformalized estimates, allowing for more confident and dependable decision-making in intervention policies.

The COP is a prediction set containing the actual outcome with a confidence level of $1-\alpha$, as detailed in Section 4.3.2 and shown in Eq. 4.3. It takes four forms, but for simplicity, it is encoded as either 0 or 1. A value of 1 means a high need for intervening when the COP set retains only the negative outcome ($COP = \{0\}$), while a value of 0 reflects uncertainty about the negative outcome, indicating less need and representing the other three forms of the COP. For example, if $COP = 0$ (only the negative outcome), the need is 1; otherwise, it is 0.

Similarly, the *CIW* provides a prediction interval, as described in Eq. 5.2, with lower (WT_{lb}) and upper (WT_{ub}) bounds for the time remaining until a negative outcome occurs. This interval offers a range rather than a single estimate, as shown in Fig. 10, reducing the risk of inaccurate timing in interventions. For example, if a negative event is predicted in four days but occurs in three, the intervention might be delayed, leading to a potential case loss. *CIW*, by offering a confidence-guaranteed interval (e.g., [2.5, 3.5] days), helps improve the timing of interventions.

$$CIW_i = [WT_{lb_i}, WT_{ub_i}] \tag{5.2}$$

The *CIE* provides a prediction interval for the intervention effect, as shown in Eq. 5.3. Rather than relying on a single estimate, *CIE* offers an interval with lower (IE_{lb}) and upper (IE_{ub}) bounds. This accounts for uncertainty in the causal effects. For example, when evaluating the impact of sending a second loan offer, if the *IE* is positive but the *CIE* ranges from -1 to 0.5, the intervention may not be advisable since the negative bound suggests potential risks.

$$CIE_i = [IE_{lb_i}, IE_{ub_i}] \tag{5.3}$$

Through this calibration step, an intervention policy is equipped with more reliable and robust estimates, enabling it to make better-informed decisions considering uncertainty measures, leading to more effective interventions and optimized resource allocation.

5.4. Online phase of the WB-PrPM

In the online phase, we operationalize the WB-PrPM approach and specify an intervention policy under resource constraints, consisting of three main components: *a filtering policy*, *a ranking policy*, and *a resource allocator*.

5.4.1. Filtering Policy

At this step, ongoing cases are continuously monitored to determine whether an intervention should be triggered based on the combined values of intervention policy parameters, i.e., *COP*, *TU*, *CIE*, and *CIW*. We assume that cases arrive in a continuous stream, and a *prefix collator* accumulates sequences of events. This process results in one or multiple ongoing cases at any given time, requiring decisions about triggering interventions while considering the dimensions of importance, urgency, uncertainty, and capacity.

The filtering policy, as discussed in Section 4.4.2, determines whether an intervention is appropriate for a given case. This policy uses a Boolean function to categorize a case as either "suitable" or "not suitable" for intervention based on predefined decision rules. In Section 4.4.2, the filtering policy considered only the need for an intervention, with suitable cases being immediately targeted for intervention. However, here in this phase, where resources are limited and additional factors influence intervention decisions, the decision rules are extended to include the dimensions of importance, urgency, and uncertainty.

Once cases are filtered and identified as candidates eligible for intervention, they must go through two additional components: the ranking policy and the resource allocator. These components determine which case is the best candidate for intervention and whether the resources are available to execute the intervention.

5.4.2. Ranking Policy

In the context of limited resources, treating all candidate cases, or even a substantial number of them, is unfeasible. Therefore, a ranking policy is designed to prioritize and rank cases needing intervention, i.e., candidate cases, to select the most suitable among them based on specific criteria.

Assuming resources are available, the goal is to select the best case among the candidates to allocate the available resources effectively. The ranking policy in the WB-PrPM approach defines the best case as the one with the highest importance, the most urgency, and the least uncertainty. This policy utilizes a linear scoring function that integrates the dimensions of importance, urgency, and uncertainty. Each dimension is weighted, allowing business analysts to assign more significance to a particular dimension according to their needs. Consequently, the ranking policy assigns a score to each candidate case, as shown in Eq. 5.4.

$$S = W_{imp} * importance - W_{urg} * urgency - W_{uncer} * uncertainty \tag{5.4}$$

In Eq. 5.4, importance is defined by the need (COP) and the effect (CIE) of the intervention. The COP, as previously discussed, can take four forms, but for simplicity, it is encoded as either 0 or 1. A value of 1 represents a high need when the COP set includes only the negative outcome ($COP = \{0\}$), while a value of 0 reflects uncertainty about the negative outcome, indicating less need and representing the other three forms of the COP. For example, if the COP set contains only the negative outcome prediction ($Y = 0$), i.e., $COP = \{0\}$, the need is 1; otherwise, it is 0. The effect is calculated by averaging the upper and lower bounds of the CIE interval. These two values—need and effect—are combined to represent the overall importance, with each weighted accordingly: W_{need} for need and W_{effect} for effect.

Similarly, the Urgency in Eq. 5.4 is derived from the Conformalized Intervention Window (CIW). The urgency value is obtained by averaging the upper and lower bounds of the CIW interval and is weighted with W_{urg}. It's important to remember that a lower urgency value indicates higher urgency because it represents the estimated time remaining until the negative outcome occurs. For instance, if one case is expected to end negatively in 4 days and another in 7 days, the first one is more urgent. Meanwhile, the uncertainty stems from the total uncertainty (TU) associated with outcome prediction and is weighted with $Wuncer$.

Thus, the scoring method incorporates all factors that influence intervention policies, ensuring that limited resources are allocated to the case with the highest score and considered the best intervention candidate.

Table 11. An example of filtering and ranking policies

caseID	*TU*	*COP*	*CIE*	*CIW*	S
A	0.2	1	0.8	0.5	1.1
B	0.5	0	0.3	2	-
C	0.6	1	0.1	3	-
D	0.7	0	0.7	0.5	-
E	0.3	0	0.1	1	-
F	0.4	1	0.6	1	0.2

Table 11 illustrates how the filtering and ranking policies can be applied to a set of ongoing cases. Let's consider six ongoing cases (labeled A to F). The business analyst establishes the following decision rule for filtering cases: If the COP=1 and the CIE is above 0.5, then the cases are filtered into candidates eligible for intervention. According to this decision rule, only ongoing Cases A and F are filtered and considered candidates for intervention. Next, the ranking policy is applied to determine which of these two cases should receive the intervention. For simplicity, we assume equal weights for all dimensions (importance, urgency, and uncertainty) in the ranking policy. Accordingly, Case A appears as the best candidate because it has a higher ranking score compared to Case F. This higher score indicates that Case A is deemed more suitable for intervention, i.e., most important, most urgent, and least uncertain. Therefore, the intervention would be applied to Case A over Case F.

5.4.3. Capacity: Resource Allocator

The *resource allocator* in the WB-PrPM framework manages the capacity of limited resources when interventions are required for multiple ongoing cases, handling the *capacity* dimension. This component is responsible for ensuring that the available resources are allocated efficiently to maximize their impact (cf. bottom-left corner of Fig. 18).

After the filtering policy identifies candidate cases suitable for intervention and the ranking policy assigns scores to these cases, the resource allocator comes into play. Given the constraint of limited resources, triggering interventions for all candidate cases is not feasible. Instead, the resource allocator continuously monitors the availability of resources within what we call the intervention resource pool. These resources are responsible for carrying out the necessary interventions.

When the ranking policy defines the best case at any given time during process execution, the resource allocator checks if there is an available resource in the intervention resource pool. If available, it is assigned to the selected case. During this allocation, the resource is "blocked" for a specified intervention duration, which is the time required to execute the intervention. Once this intervention duration has elapsed, the resource is "released" and returned to the intervention resource pool, making it available for other cases that may require intervention.

To illustrate this, consider the example presented in Table 11, where six ongoing cases are filtered into two candidate cases, Cases A and F. Suppose we have only one available resource in the intervention resource pool. The resource allocator would allocate this resource to Case A first and block it for the required intervention duration. After this duration passes, if Case F still meets the filtering decision rule and has the highest ranking score, the now-available resource will be allocated to Case F.

This process of monitoring, allocating, blocking, and releasing resources ensures that interventions are applied to the best cases while optimizing the use of the limited resources available.

5.5. Evaluation

In this section, we instantiate the WB-PrPM framework to construct several internal variants and evaluate their performance across different levels of resource utilization. This evaluation examines the internal variants of the WB-PrPM. It compares their effectiveness to existing baseline methods [202, 341, 251], which do not account for resource constraints or the inherent uncertainties within the intervention policy. Thus, and to guide the evaluation, we break down RQ2 into two sub-questions:

RQ2.1: How do different variants of the WB-PrPM approach compare against each other in terms of total gain across varying levels of resource utilization?

RQ2.2: To what extent does the automated optimization of the approach's parameters lead to intervention policies with higher total gain than non-optimized ones (i.e., variants of the framework with default parameter settings)?

The specific research questions outlined above aim to provide insights into how business stakeholders can effectively specify and optimize intervention policies within the WB-PrPM framework while addressing the challenges posed by resource constraints. These questions demonstrate the flexibility and configurability of intervention policies based on organizational needs. For instance, one organization may prefer to specify an intervention policy using only predictive and conformal estimates, while another may wish to base it on causal measures. To illustrate this flexibility, we explore possible variants of the WB-PrPM in response to RQ2.1. In contrast, RQ2.2 focuses on enabling stakeholders to automate the parameters of intervention policies, thereby relieving analysts of the burden of determining these parameters. It emphasizes that stakeholders not only need to define intervention policies but also have the opportunity to enhance these policies through automated optimization, which is essential for effective resource management in real-world scenarios.

Accordingly, this leads us to conduct an ablation study. An ablation study is a method used to evaluate machine learning methods by sequentially removing components of the method to understand each component's contribution to overall performance. In our context, the components are the models used to estimate the factors of an intervention policy—specifically, the need (COP and TU), effect (CIE), intervention window (CIW), and intervention effect (CIE). All components are included in a maximalist variant of the approach, and capacity constraints are explicitly considered. We then remove each component to obtain different variants.

5.5.1. Datasets

To evaluate the WB-PrPM approach, we utilized two publicly available event logs, BPIC2012 and BPIC2017, representing the loan origination process. These logs were also used to evaluate the UN-PrPM approach discussed in Chapter 4 and are explained in detail in Section 4.2.1. Here, we briefly describe them.

Each case's outcome is determined by whether the loan application concludes successfully, based on a specific condition (boolean function) evaluated on a completed case. Outcomes are defined according to each case's final activity. If the last activity indicates a negative outcome, the proposed intervention is to send an additional loan offer to the applicant. This intervention is identified by the "Send_Offer" activity, which aims to increase the likelihood of the applicant accepting a loan by presenting an additional offer. We categorize cases based on the number of offers sent: cases that receive only one offer are placed in the control group ($T = 0$), while those that receive more than one offer are placed in the treatment group ($T = 1$).

5.5.2. Experimental Setup

As discussed in Section 4.5.2, we utilized Python version 3.10 for our experiments. A temporal split approach is employed to divide each log into three subsets: training (50%), validation (D_{val}: 25%), and testing (D_{test}: 25%). Cases are sorted chronologically by their timestamps to ensure that the model is trained and tested in a realistic scenario where future cases are predicted based on past data. Additionally, the training set is further divided to create a calibration set, resulting in the following splits: training set (D_{train}: 40%) and calibration set (D_{cal}: 10%).

The training set (D_{train}) is used to train the predictive, causal, and survival models. The calibration set (D_{cal}) is utilized to derive conformalized estimates, such as the Conformalized Outcome Prediction (COP), Conformalized Intervention Effect (CIE), and Conformalized Intervention Window (CIW). The validation set (D_{val}) is employed to autotune the parameters of the filtering and ranking policies, specifically focusing on optimizing the thresholds and weights. Finally, the test set (D_{test}) is used to measure the total gain, which is the key metric reported in the evaluation results.

As discussed in Section 4.5.2, we used an ensemble approach [203] for the predictive model, specifically leveraging CatBoost [22]. This model is trained to estimate the probability of a negative outcome, denoted as OP, and the total uncertainty (TU) associated with the outcome prediction. In parallel, the causal model is trained using the Orthogonal Random Forest (ORF) algorithm from the *EconML*[1] library to estimate the intervention effect (IE). ORF is chosen for its ability to handle high-dimensional data, such as event logs.

The survival model is trained using the Cox proportional hazards method [34], a widely adopted statistical model in survival analysis. This method investigates the association between covariates (independent variables) and the hazard rate, which denotes the risk of an event occurring over time. To implement this method, we used the *lifelines*[2] Python library, specifically designed for survival analysis, which provides a toolkit to enhance the precision and reliability of our survival estimates.

Regarding the conformal prediction model, which converts point estimates (OP, IE, and IW) into a conformalized prediction set (COP) or interval (CIE and CIW), we followed the inductive conformal prediction approach. We used the Python package *ACPI*[3] to derive COP and CIW. This package is well-suited for both classification and regression machine learning tasks, enabling it to convert the point estimates into prediction sets or intervals that contain the true value with a predefined confidence level. On the other hand, for deriving conformalized intervention effects, we used the *cf-causal*[4] library, a conformal inference tool developed in the R programming language. This library is specifically designed to obtain conformalized causal estimates, making it an ideal choice for our needs.

Concerning the ranking policy, the experiments investigate scenarios with equal weights and varying weightings for each dimension. We classify weights into three levels: *low*, *medium*, and *high* $(0.5, 1, 2)$, offering a range of options to tailor the ranking policy to different settings.

5.5.3. Resource Utilization

Resource utilization (ρ) measures resource allocation efficiency in the WB-PrPM approach, assessing intervention policy effectiveness under varying constraints. ρ is derived from the demand-to-capacity ratio (Eq. 5.5). *Demand* is the total resource requirement during testing, calculated as the number of triggered interventions multiplied by the average intervention duration. *Capacity* is the total resource potential, given by available resources multiplied by execution time. The ratio quantifies resource usage for interventions.

$$\rho = \frac{Demand}{Capacity} \tag{5.5}$$

---

[1] `https://github.com/microsoft/EconML`

[2] `https://github.com/CamDavidsonPilon/lifelines/tree/master`

[3] `https://github.com/salimamoukou/ACPI`

[4] `https://lihualei71.github.io/cfcausal/reference/conformalIte.html`

To assess the effectiveness of the WB-PrPM approach under diverse resource constraints, we explored four levels of resource utilization: *high*, *moderately high*, *medium*, and *low*. These levels represent varying resource availability and demand, allowing for the evaluation of the approach in different operational scenarios.

- *High Resource Utilization:* This level corresponds to scenarios with severely limited resources, such as only one available loan officer. In such cases, the demand for interventions exceeds the available resources, leading to high ρ values. This level tests the approach's ability to manage resources effectively under extreme scarcity.
- *Moderately High Resource Utilization*: This level represents situations where resources are available but not fully exploited. It provides insights into how well the approach can optimize resources when constrained but not at maximum capacity.
- *Medium Resource Utilization:* This level reflects typical real-world resource management conditions. It provides a baseline for performance evaluation, offering a standard against which other levels can be compared.
- *Low Resource Utilization:* At this level, multiple resources are available to execute interventions at each decision point. However, the decision may be to refrain from utilizing them when no incremental gain is achieved by executing the intervention for a given case. This scenario examines the approach's ability to allocate resources when they are more available.

Table 12. Extraction of the resource utilization (ρ) levels across various number of available resources (n).

				Resource utilization (ρ)			
Event log	# Triggered interventions	I_{dur} (s)	Duration of the log (s)	High $\rho \geq 90\%$	Moderately High $90\% > \rho \geq 75\%$	Medium $75\% > \rho \geq 50\%$	Low $50\% > \rho \geq 25\%$
BPIC2012	1172	1	365	$\mathbf{1} \leq n \leq 3$	$3 < n \leq \mathbf{4}$	$4 < n \leq \mathbf{6}$	$6 < n \leq \mathbf{12}$
BPIC2017	7852	1	3630	$\mathbf{1} \leq n \leq 2$	$2 < n \leq \mathbf{3}$	$3 < n \leq \mathbf{4}$	$4 < n \leq \mathbf{8}$

The resource utilization levels are determined post-operation for each log. We first ran experiments with varying numbers of available resources. After these experiments, we analyzed the data to establish the resource utilization levels. Specifically, we recorded the log execution duration during the testing phase, the number of interventions triggered during that duration, and the average intervention duration, as detailed in Table 12. By varying the available resources and calculating ρ for each scenario, we could evaluate how the WB-PrPM approach performs under different resource constraints, allowing us to establish the resource utilization levels for our analysis.

5.5.4. Variants of the WB-PrPM approach

This section explores various variants of the WB-PrPM framework through an ablation study approach. We consider different configurations by including or excluding specific factors, such as need, effect, etc. The primary distinction among these variants lies in the structure of the decision rules employed in the filtering process. Additionally, we incorporate several baseline models for comparative analysis. The following describes each variant and its approach to developing an intervention policy (IP):

- **IP1 (Predictive):** This variant relies solely on the outcome prediction (OP) estimates to guide interventions. Interventions are triggered when the OP exceeds a predefined threshold (τ_{OP}). This threshold is set by default to 0.5, following a similar approach used in probabilistic classification machine learning models. However, the threshold can vary between 0.1 and 0.9, depending on the specific conditions of the intervention policy. This variant serves as a baseline, imported from previous literature, according to the work done in [202].
- **IP2 (Causal):** This variant uses causal relationships to filter cases based on the intervention effect (IE). Interventions are triggered when the IE exceeds a threshold (τ_{IE}), set by default to 0, meaning the intervention is expected to have a positive impact. The threshold can be adjusted empirically between 0.0 and 1, with 1 representing the maximum effect. This variant is a baseline from previous literature based on the work in [285].
- **IP3 (Reliability):** This variant uses a *reliability estimate* (RE), introduced in [251] for black-box intervention policies. A typical example is the class probability from an ensemble classifier, where RE reflects the fraction of base learners predicting a positive or negative outcome. Cases become candidates for intervention when the RE exceeds a threshold (τ_{RE}), set by default to 0 but adjustable between 0 and 1. This variant also serves as a baseline.
- **IP4 (Remaining Time):** This variant estimates the remaining time (RT) until case completion, irrespective of whether a negative outcome occurs. The concept of the remaining time is drawn from predictive process monitoring literature and serves as a baseline to compare against methods that estimate the time until a negative outcome, addressing the urgency of interventions. Cases are filtered as candidates when their RT is below a set threshold (τ_{RT}) (e.g., ten days), which can vary from 1 day up to the mean value derived from the validation set.
- **IP5 (Urgency):** This variant is similar to IP4 in that it prioritizes urgent cases by assessing the time left before case completion in IP4 or until a negative outcome occurs in IP5. Ongoing cases are filtered based on the intervention window (IW). When the IW falls below a certain threshold (τ_{IW})—set by default based on the minimum number of days from the validation set—cases are classified as urgent and require intervention. The IW can vary from one day up to the mean value derived from the validation set.
- **IP6 (Total Uncertainty):** This variant incorporates total uncertainty (TU) measures into the decision-making process. Ongoing cases are filtered as

candidates when the TU falls below a specified threshold (τ_{TU}), with the default value set at 0.5. The threshold can vary from 0 (indicating low uncertainty) to 1 (indicating high uncertainty).

- **IP7 (Predictive + Conformal):** This variant combines outcome prediction (OP) with conformalized outcome prediction (COP), filtering cases where $OP > 0.5$ and $COP = 1$. Intuitively, the decision rule "$OP > 0.5$ and $COP = 1$" indicates that the outcome prediction for an ongoing case is above the threshold of 0.5, suggesting a higher likelihood of a negative outcome. A COP of 1 indicates that, given a predefined user significance level (e.g., $\alpha = 0.1$), the model is confident that the case will end negatively. For instance, with $\alpha = 0.1$, the model is 90% confident ($1-\alpha$) that the outcome will be negative. The user-defined significance level (α) sets the certainty for the prediction; a lower significance level (e.g., $\alpha = 0.1$) means the COP set will include only the negative outcome ($Y = 0$), reflecting high confidence in the prediction. This approach filters out cases with higher uncertainty, focusing on those with more certain negative outcomes.
- **IP8 (Causal + Conformal):** This variant extends the causal-based policy (IP2) by incorporating conformal prediction. In addition to filtering ongoing cases as candidates based on the intervention effect (IE) above a threshold (τ_{IE}), we utilize the conformalized intervention effect (CIE), which provides a prediction interval with lower and upper bounds for the intervention effect. The CIE value is calculated by averaging the lower and upper bounds, with a default threshold ((τ_{CIE})) set to zero. This value can vary between zero and the maximum upper bound derived from the validation set for empirical thresholding. Additionally, the range of the prediction interval is controlled by the confidence level ($1-\alpha$), which the user determines, similar to IP7. The default confidence level is set at 0.5 but empirically varied between 0.1 and 0.9.
- **IP9 (Remaining Time + Conformal):** This variant combines the remaining time (RT) estimate with conformal prediction. It filters ongoing cases based on the remaining time until case completion and the conformalized remaining time (CRT) estimate, which is calculated as the average of the interval's upper and lower bounds. Cases are classified as candidates when both the RT and the CRT fall below a specified threshold (τ_{RT}) (e.g., five days). This threshold can vary based on empirical observations from the validation set, allowing for dynamic adjustments based on historical data. Similarly, the CRT interval lower and upper bounds are based on the defined confidence level ($1-\alpha$).
- **IP10 (Urgency + Conformal):** This variant resembles IP9 but uses the intervention window (IW) and the conformal interval window (CIW) estimates derived from the survival model instead of the RT and CRT.

- **IP11 (Predictive + Conformal + Total Uncertainty):** This variant integrates OP, COP, and TU measures to establish a more conservative intervention policy.
- **IP12 (Predictive + Conformal + Causal + Conformal):** This policy combines parameters of IP7 and IP8, employing predictive and causal decision rules along with conformal prediction to filter cases based on a mixture of outcome predictions and causal inference under high confidence.
- **IP13 (Predictive + Conformal + Urgency + Conformal):** This variant combines IP7 and IP10, incorporating predictive, urgency, and conformal rules to ensure that cases requiring urgent interventions are addressed with high predictive certainty.
- **IP14 (Causal + Conformal + Urgency + Conformal):** This variant integrates IP8 and IP10, focusing on urgent cases driven by causal relationships while maintaining high confidence through conformal rules.
- **IP15 (Predictive + Conformal + Causal + Conformal + Urgency + Conformal):** This is the most conservative policy, merging predictive, causal, urgency, and conformal decision rules to maximize intervention effectiveness while considering resource constraints. It combines elements from IP7, IP8, and IP10.

The variants described above represent just a few of the many configurations that can be created using the WB-PrPM framework. These examples are not exhaustive. For instance, an alternative variant could use only conformalized intervention effects or a conformalized intervention window without combining them with point estimates such as the intervention effect or intervention window, as demonstrated in IP8 and IP10. This flexibility enables organizations to adapt and customize intervention policies based on their specific requirements.

Each variant also involves parameters and thresholds that need to be defined. For instance, in IP1, the parameter is the outcome prediction (OP), and the threshold is τ_{OP}. This highlights the importance of automated methods to optimize these threshold values in intervention policies to improve particular performance measures, such as total gain.

5.5.5. Results

We present the results of evaluating the WB-PrPM framework for developing white-box intervention policies, considering key dimensions such as importance, urgency, uncertainty, and capacity. The evaluation examines different framework variants under varying levels of resource availability, focusing on identifying an appropriate performance measure to assess the effectiveness of these intervention policies.

Each intervention policy (e.g., IP1, IP2, ..., IP15) aims to target cases where, without intervention, the outcome would be negative, but with intervention, the outcome becomes positive. We refer to these cases as "persuadable cases," as illustrated in Fig. 12. Specifically, these are cases where the outcome without intervention is $Y(T=0)=0$, and with intervention, it changes to $Y(T=1)=1$. The goal is to avoid intervening in cases classified as "do not disturb," "sure thing," or "lost causes," as there would be no benefit from intervention in these cases. Moreover, intervening in such cases could have negative effects, leading to resource waste and potential loss.

The effectiveness of intervention policies is measured by the $TotalGain$, as discussed in 4.5.2 and as shown in Eq. 4.4, which incorporates the change in outcome (ΔY). ΔY can take a value of either zero or one: a value of one indicates a successful intervention, where the outcome changes from negative to positive. In contrast, a value of zero indicates no change in outcome. Additionally, the total gain function considers two other factors: the benefit of achieving a positive outcome ($Gain$) and the cost of intervention (C_{in}). While beneficial, interventions incur costs, which are necessarily factored into the total gain, as shown in Eq. 4.4.

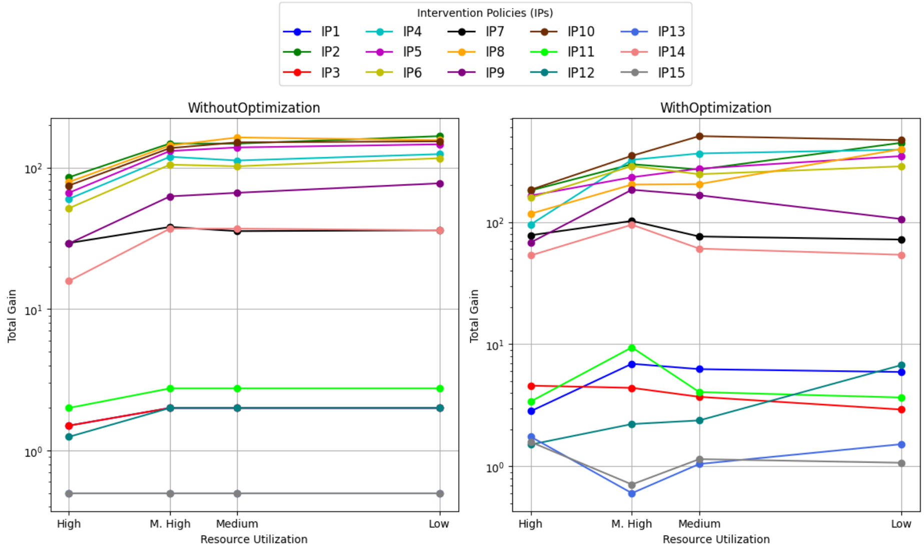


Figure 19. *BPIC2012*: Total Gain Comparison of All Variants

In response to RQ2.1, we perform a comparative analysis of all intervention policy variants, including the baseline policies (IP1, IP2, IP3, and IP4), under various resource utilization levels (High, Medium-High, Medium, and Low). The results of this analysis are presented in Figures 7 for the BPIC2012 dataset and Figure 8 for the BPIC2017 dataset. These figures also illustrate the differences between applying hyperparameter optimization methods (figure on the right) and using default parameter threshold values (figure on the left) for each intervention policy, thereby addressing RQ2.2.

Here, we observe three distinct behaviors. The first observation is that simpler and less conservative variants, which include only one or two parameters in the decision rules used to filter ongoing cases into candidates eligible for intervention, tend to outperform more complex variants that combine multiple parameters. Compared to the more complex or conservative variants, these simpler variants demonstrate better performance due to their streamlined structure and fewer decision-making factors. This trend is evident across both event logs, particularly for IP2, IP8, IP10, and IP5 variants. These policies incorporate the dimensions of importance, represented by the intervention effect factor, and urgency, represented by the intervention window, allowing them to achieve more effective outcomes.

The second observation is that incorporating uncertainty or conformal prediction measures on top of simple estimates derived from machine learning models, e.g., predictive, enhances performance. This improvement is clearly seen when comparing IP3 with its conformal variant, IP9, where the RT estimate is combined with the CRT estimate. Similarly, IP6, which relies primarily on the total uncertainty measure of the outcome prediction, shows better performance. This trend is also evident when comparing IP1 against its conformal variant, IP7. In this case, adding conformal measures in addition to the point estimate of the outcome prediction significantly improves policy performance. Intuitively, this can be explained by the fact that relying solely on point estimates, which inherently carry uncertainty, may not be ideal for decision-making processes unless these estimates are conformalized to account for their uncertainty.

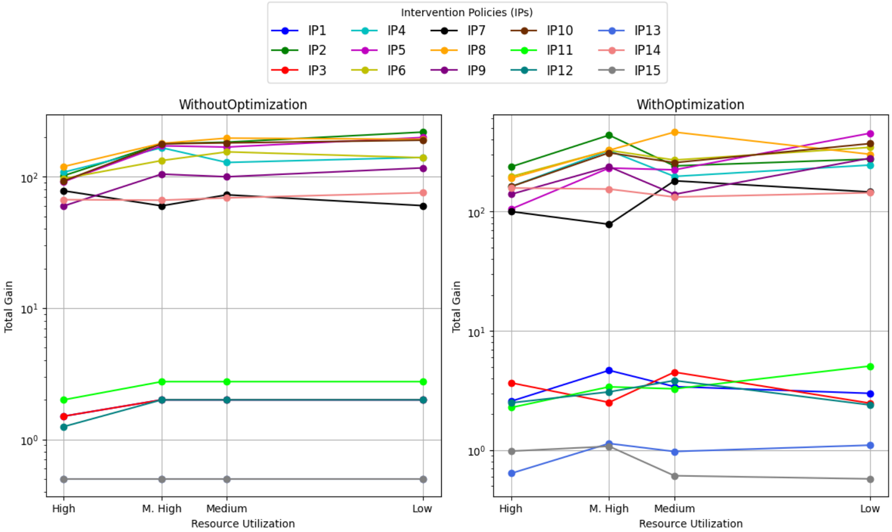


Figure 20. *BPIC2017*: Total Gain Comparison of All Variants

The third observation is that complex variants with multiple parameters may underperform. Variants with more complex decision rules—those that include three or more parameters—tend to underperform. This underperformance is likely due to the limited availability of historical data matching such decision rules' complex structure. This is particularly evident in variants such as IP11, IP12, IP13, and IP15, each of which has at least three parameters, meaning three conditions must be simultaneously met. However, this observation does not hold in the case of IP14, which combines intervention effect and intervention window (representing causal and urgency) and conformalized estimates. In the context of business processes, where cases may overlap or execute in parallel, prioritizing cases and knowing the expected impact of interventions could prove useful, especially for resource-constrained environments. Though IP14 outperforms other complex variants, it still does not surpass the performance of simpler variants.

Based on these observations, when more conservative policies are needed, focusing on factors like intervention effect and/or intervention window, coupled with a layer of uncertainty, is advisable to ensure confident decision-making. For example, in the banking industry, a conservative approach is necessary to minimize default rates when deciding on high-risk loans. The bank may evaluate the intervention effect (how offering a loan could change a customer's financial outcome) and the intervention window (timing of the loan approval). By incorporating uncertainty estimates into the decision-making process, the bank can ensure that the decision to approve or reject a loan is based on more reliable predictions, thus avoiding unnecessary financial risk. For instance, offering loans to customers with ambiguous credit histories could be managed more effectively with uncertainty-aware policies.

Conversely, simpler variants tend to outperform more complex ones in situations where more relaxed policies are preferred. For example, in e-commerce, customer retention strategies often involve simple, flexible intervention policies. For example, to target cart abandonment, a company may implement a simple intervention policy that triggers a discount offer when customers leave items in their cart for a certain time (urgency dimension).

To address RQ2.2, we utilized the Tree-structured Parzen Estimator (TPE) algorithm [297]. We conducted 50 optimization trials to fine-tune parameters such as filtering thresholds and ranking weights for each intervention policy variant. The main objective was to examine the tradeoffs between different parameter values within the intervention policies, enabling optimization for enhanced performance. This approach assists business analysts in identifying the optimal combinations of intervention policy parameters rather than manually setting them.

As illustrated in Figures 19 and 20 (on the right), the parameter optimization process improved total gain compared to non-optimized default parameters. This improvement was consistently observed across all policy variants, including baseline methods, highlighting the effectiveness of the optimization in enhancing intervention policy outcomes in both datasets.

An important observation from the parameter optimization analysis is that as the availability of resources increases, the execution times for determining the optimal values for different parameters decrease. This increased efficiency is attributed to the higher resource availability, facilitating a more effective optimization process. No clear pattern emerged regarding ranking policy weight values; their variations appeared to depend more on the level of available resources and the type of variant used.

Also, the analysis reveals that when parameter optimization is not applied, most variants achieve higher total gain as resource utilization decreases, indicating more available resources for intervention. This suggests that without optimal settings, the framework allows for more aggressive strategies, capturing a wider range of cases.

In contrast, with parameter optimization, most variants plateau in performance at medium resource levels. This means that the decision rules become more conservative, narrowing the range of cases targeted for intervention. Stricter filtering criteria reduce the number of interventions, limiting total gain.

This behavior reflects a trade-off between resource allocation and maximizing positive outcomes. When resources are constrained, the system focuses on interventions likely to provide the highest benefits, relying more on conformal estimates to assess uncertainty. However, with adequate resources, the framework can adopt a more flexible approach, increasing the chances of identifying cases that would benefit from intervention.

For the causal variant (IP2), the intervention effect threshold remained unchanged from the default value. However, when the causal variant was combined with conformal prediction (IP8), the average value of the intervention effect (IE) threshold increased to 0.3, particularly at low or medium levels of resource availability, with the causal intervention effect (CIE) threshold around 0.5. Additionally, we observed that as the level of available resources increased, the causal intervention window (CIW) threshold also rose in the IP10 variant. This trend can be explained by the fact that increased resources allow for a longer time frame to execute interventions.

5.5.6. Qualitative Analysis of Interpretable Patterns

To complement the quantitative evaluation of intervention policies (IP1–IP15), we analyze the interpretable decision rules derived from the WB-PrPM framework. These rules highlight how transparency in white-box models enhances policy design. Below, we dissect three key patterns and their implications:

1. *Pattern 1: Simpler Rules Outperform Complex Ones.* Intervention policies, including importance (intervention effect) and/or urgency (intervention window) as decision criteria (e.g., IP2, IP5, IP8, IP10) consistently outperformed complex variants with >3 parameters (e.g., IP11–IP15). For instance, IP8's rule—"Intervene if intervention effect > 0.3 and the confidence level equal to 90%"—achieves high TotalGain by focusing on cases where effective action maximizes impact. From a policy perspective, simple thresholds enable stakeholders to prioritize cases with clear cost-benefit tradeoffs, optimizing resource allocation. For example, in e-commerce, urgency-based rules, such as "Offer discounts if cart abandonment risk > 70% within 1 hour," align with dynamic customer behavior, ensuring timely interventions.

2. *Pattern 2: Uncertainty-Aware Policies Enhance Confidence.* Policies that integrate uncertainty measures, such as confidence levels or total uncertainty thresholds, consistently outperform those that do not. For instance, IP8's rule—"Intervene if the intervention effect is greater than 0.2 and the confidence level is 80% ($\alpha = 0.2$)"—helps counteract overconfidence in noisy predictions by ensuring that interventions are only applied when strongly supported. From a risk mitigation perspective, confidence level thresholds (e.g., "$\alpha = 0.2$") can quantify uncertainty, enabling more careful policies in high-stakes areas like healthcare or finance. Moreover, stakeholder trust is supported when decision-making frameworks include clear uncertainty measures. In pilot studies, clinicians preferred policies like IP7, which incorporated confidence level (e.g., "70% confidence in treatment success") over models with ambiguous predictions.
3. *Pattern 3: Overly Complex Rules Suffer from Data Sparsity.* Multi-parameter rules (e.g., IP12: "$IE > 0.4$, $\alpha = 0.2$, $OP > 0.6$, $\alpha = 0.3$") are designed to provide more conservative policies, but they often underperform due to limited historical data that meets all the specified criteria. As a result, only a small number of cases satisfy the stringent thresholds, leading to missed opportunities for intervention. This highlights the importance of balanced policy design, as using more than two concurrent thresholds can be impractical unless supported by sufficient data or required for a stricter policy. Furthermore, the success of IP8 ("$IE > 0.3$ and $\alpha = 0.1$") illustrates that combining causal and conformal measures can compensate for the complexity of overly intricate rules. This approach allows for a more adaptable strategy, which remains effective despite data sparsity, striking a better balance between complexity and actionable insights while reducing the risk of missing valuable interventions.

5.5.7. Discussion and Threats to Validity

In this section, we discuss the implications of our findings, emphasizing the practical challenges in implementing the WB-PrPM framework and identifying potential threats to the validity of our results. We also examine the suitability of different intervention policy variants based on specific process characteristics and highlight practical limitations that may hinder users from effectively utilizing the framework.

Practical Challenges and Change Management. The WB-PrPM framework offers a flexible approach for stakeholders to define intervention policies tailored to their business needs. However, its practical application comes with several challenges. One major issue is the change in how business processes are managed. Automated intervention policies can disrupt established workflows and may meet resistance from employees and stakeholders who are accustomed to making decisions manually. This resistance is particularly strong in organizations where decisions have traditionally relied on human intuition or experience.

To address this challenge, it is essential to implement a method for estimating the confidence of each decision. This transparency can help stakeholders understand and trust the system's recommendations. By showing users how certain the system is about a specific intervention, they may feel more comfortable delegating decision-making to the framework. Additionally, mechanisms that explain how the system arrives at its decisions are vital for building trust and facilitating the transition from manual to automated processes.

Integrating the WB-PrPM framework with existing business process management systems can also present challenges. Legacy systems may only partially support real-time monitoring of interventions, and organizations might incur significant costs to adapt or replace these systems. Beyond technical issues, the change management process must consider the organizational culture and ensure alignment among all stakeholders with the system's objectives. Training and ongoing support will be necessary for users to understand and effectively engage with the framework.

Applicability Across Process Contexts. While the WB-PrPM framework provides a clear method for developing intervention policies, the effectiveness of various variants heavily depends on the characteristics of the processes involved, such as resource availability, cost structures, and sensitivity to outcomes. More complex variants, especially those that combine multiple models, may not perform well in environments requiring flexibility due to their conservative nature. In contrast, conservative policies could be advantageous in processes where negative outcomes carry a high cost, as they prioritize only the most promising interventions, reducing the risk of misallocating resources. For example, in the banking industry, a conservative approach is necessary to minimize default rates when deciding on high-risk loans.

In processes with high variability or uncertainty in outcomes, simpler variants may be more effective because they can adapt better to changing conditions. Organizations facing resource constraints may also benefit from these simpler policies, as they balance performance and efficient resource allocation. Therefore, we recommend that organizations carefully evaluate their process characteristics before selecting an intervention policy. For example, to target cart abandonment, a company may implement a simple intervention policy that triggers a discount offer when customers leave items in their cart for a certain time (urgency dimension).

Flexibility of Models. The WB-PrPM framework allows the integration of various machine learning and statistical models, enhancing flexibility in making predictions and decisions based on specific process characteristics and available data. Alternatives such as Propensity Score Matching (PSM) and Inverse Probability of Treatment Weighting (IPTW) can complement the potential outcome framework by balancing features between treated and untreated groups to estimate the intervention effect [309].

To address uncertainty in estimates related to need, effect, and intervention timing, Monte Carlo Simulation can be utilized to simulate multiple possible outcomes for a case and assess uncertainty in predicted intervention results [310]. Moreover, rather than using survival models to estimate the time until a negative outcome occurs, methods focusing on remaining-time prediction can be employed, which incorporate confidence measures for greater accuracy.

A wide range of classification methods, including Random Forest and Logistic Regression, can be applied to outcome-oriented prediction tasks. Additionally, Queuing Theory can be utilized to estimate resource availability for interventions in processes with limited capacity, enabling effective prioritization of interventions [312]. Furthermore, Learn-to-Rank (LTR) methods can enhance the ranking of cases based on intervention criteria, optimizing decision-making within the approach [311].

Threats to Validity. The WB-PrPM approach shares similar external validity concerns with the UN-PrPM approach, primarily due to the reliance on only two datasets, which limits the generalizability of the evaluation. Another threat to external validity stems from the use of the RealCause [295] method to estimate alternative outcomes, as it extrapolates what the outcome would have been if an intervention had not occurred. These external validity threats are discussed in detail in Section 4.5.4. Beyond these external validity concerns, the WB-PrPM approach also faces threats to internal and ecological validity.

Internal Validity. Potential threats to internal validity may arise from the resource allocation procedure during runtime. To address this potential threat, we conducted each experiment three times to account for uncertainties and reported the mean value across the replications. Despite minor variations in total gains, we consistently obtained similar results across these replications.

Ecological Validity. Potential threats to the ecological validity of the findings stem from the assumption that all resources have uniform proficiency in executing interventions. Furthermore, the proposed framework assumes the existence of only one type of intervention. In reality, there might be various types of interventions, such as contacting a customer to offer a discount or providing a personalized consultation. The proposed framework is not tailored for scenarios involving multiple types of interventions, commonly referred to as multi-intervention settings.

5.6. Summary

This Chapter introduced the White-Box Prescriptive Process Monitoring (WB-PrPM) approach, aimed at improving the transparency and effectiveness of intervention policies in business processes. While the UN-PrPM approach presented in Chapter 4 focuses on developing uncertainty-aware intervention policies, the WB-PrPM extends this by incorporating resource constraints, ensuring that policies operate effectively under limited resources. Thus, this Chapter builds upon previous work and discusses factors affecting intervention policies, categorized into four top-level dimensions: importance, urgency, uncertainty, and capacity, to develop intervention policies that operate under limited resources. This results in intervention policies ranging from more conservative to aggressive, allowing for a balance of different scenarios and benefit-to-cost tradeoffs.

Accordingly, the WB-PrPM approach integrates predictive and causal models to estimate interventions' needs and expected effects. It employs survival models to assess urgency, conformal prediction techniques to manage uncertainty, and resource capacity monitoring to incorporate resource limitations directly into decision-making. Users can define policy parameters, such as thresholds and weights, to filter ongoing cases and identify candidates for intervention. Candidate cases are ranked based on importance and urgency, ensuring interventions maximize a performance measure (total gain). Additionally, the approach automates the tuning of policy parameters to optimize the total gain achieved by the intervention policy.

A comparative analysis was conducted using real-life datasets to evaluate the performance of various WB-PrPM approach variants against established baselines. The results demonstrate that WB-PrPM consistently achieves higher total gains compared to existing baselines. Notably, incorporating causal effects of interventions and urgency as decision-making dimensions significantly enhances the prioritization of time-sensitive cases, ensuring that interventions are applied where they yield the greatest impact. The analysis also reveals that more complex variants, which include multiple decision parameters, tend to adopt a more conservative strategy compared to simpler, more relaxed variants. This adaptability proves particularly advantageous when balancing cost-to-benefit tradeoffs. Furthermore, simpler variants that integrate uncertainty measures consistently outperform those that do not, highlighting the value of uncertainty awareness in improving intervention effectiveness.

The results also show that while complex variants tend to underperform simpler variants due to their strict decision rules, a black-box setting may yield different benefits. In this context, the decision to trigger interventions and allocate limited resources becomes more data-driven, which could favor the decision-making capabilities of complex variants compared to simpler ones. This shift highlights the potential value of incorporating complexity into intervention policies to enhance overall effectiveness.

Thus, the next Chapter will further extend the UN-PrPM and WB-PrPM approaches to develop intervention policies in black-box settings and constrained resource environments.

6. PRESCRIPTIVE PROCESS MONITORING UNDER RESOURCE CONSTRAINTS: A BLACK-BOX APPROACH

In the preceding chapters, we presented two distinct approaches to prescriptive process monitoring: the uncertainty-aware prescriptive process monitoring (UN-PrPM, Chapter 4) and the white-box prescriptive process monitoring (WB-PrPM, Chapter 5). Both approaches aim to identify cases likely to result in negative outcomes during the execution of business processes and to trigger interventions that can transform these outcomes into positive ones. These interventions are designed to maximize a total gain function, with the UN-PrPM approach assuming resource-unconstrained environments, while the WB-PrPM approach operates under resource constraints. A limitation of these approaches is their dependence on manual specifications by business analysts to define the parameters that most influence the policy and determine the threshold values for triggering interventions.

The WB-PrPM approach, for instance, involves the formulation of intervention policies through a filtering and ranking process that considers various factors, including importance (need and effect), urgency (intervention window), uncertainty (as measured by conformal predictions), and capacity (resource availability). While this approach provides transparency and control, it also requires significant input and decision-making from the business analyst, who must deeply understand the process dynamics. This manual process can be time-consuming and sensitive to human error, particularly in complex scenarios where multiple factors interact unpredictably. Meanwhile, online reinforcement learning (RL) has emerged as a promising data-driven approach for developing intervention policies via trial-and-error. However, existing RL-based approaches assume that resources for interventions are unlimited—an impractical assumption in real-world settings.

In this context, this chapter addresses a specific instance of the general PrPM problem: *Given past observations of completed cases, where some cases had interventions, and each intervention has a cost and consumes resources with limited capacity—the research question is posed as follows:* ***RQ3:*** *How can we design data-driven, black-box intervention policies that manage limited resources effectively, optimizing resource allocation and process performance, i.e., total gain, within a resource-constrained setting?*

To address this question, this chapter introduces a *black-box prescriptive process monitoring (BB-PrPM)* approach as an alternative to the previously discussed methods. The BB-PrPM approach utilizes RL, specifically designed to learn from historical process data, to automatically develop intervention policies without the need for filtering and ranking steps and, accordingly, the need to explicitly specify thresholds or parameter weights.

The BB-PrPM approach mitigates the limitations of the WB-PrPM approach by abstracting the complexity of policy formulation into a data-driven model. This abstraction simplifies decision-making for business analysts, allowing them to rely on the model's predictions rather than manually setting thresholds and parameters. In this context, we introduce an RL-based approach for prescriptive process monitoring that triggers interventions based not only on the necessity, timeliness, and effect of the intervention but also on the uncertainty associated with these predictions and the level of resource utilization.

As in the WB-PrPM approach, the BB-PrPM approach uses predictive and causal models to estimate an intervention's necessity and expected effect. It also incorporates a survival model to assess the urgency of interventions and employs conformal prediction techniques to account for uncertainty. Furthermore, the approach includes a capacity-monitoring component that tracks resource allocation, utilization, and demand intensity, ensuring that these parameters are integrated into the intervention policy. The key distinction between the WB-PrPM and BB-PrPM approaches lies in the black-box nature of the latter's intervention policies.

We report on an empirical evaluation to assess the effectiveness of various BB-PrPM approach variants, particularly in terms of convergence and performance metrics across various resource utilization levels. This evaluation is compared to baseline methods that are RL-based but lack consideration for resource constraints and inherent uncertainty parameters within the policy. The objective is to demonstrate how the RL agent converges quickly toward intervention policies that maximize total gain post-convergence.

The chapter begins with an introduction to the core concepts underlying our approach in Sections 6.1 and 6.2, followed by a detailed explanation of the BB-PrPM approach in Section 6.3 and Section 6.4. The empirical evaluation is then presented in Section 6.5, and concludes with a summary in Section 6.6.

6.1. High-Level Approach Overview

The BB-PrPM approach develops a data-driven intervention policy for business process execution using an online reinforcement learning (RL) method that balances interventions' necessity with limited resource availability constraints. This approach shares the same foundational structure as the WB-PrPM method introduced in Chapter 5, encompassing three main components: (1) identifying factors influencing intervention policies, (2) an offline phase, and (3) an online phase.

The first component, *identifying factors influencing intervention policies*, is largely similar to the WB-PrPM approach, though there are key differences that require discussion. Specifically, these factors are categorized into four top-level dimensions that influence intervention decisions: Importance (need and effect), Urgency (intervention window), Capacity (resource availability), and Uncertainty (conformal measures). These dimensions guide the RL agent's decisions on whether and when to trigger an intervention and how to allocate resources to specific cases. The primary distinction in this component compared to the WB-PrPM lies in the capacity dimension, which will be detailed in the following subsection.

The *offline phase* (see Fig. 22) in the BB-PrPM approach is entirely analogous to the WB-PrPM approach (see Section 5.3), where various machine learning models are trained to predict or estimate the identified factors during operational scenarios. A predictive model estimates the need for intervention, while a causal model assesses the potential effect of the intervention—both of which pertain to the importance dimension. Additionally, a survival model is trained to estimate the remaining window for feasible intervention, representing the urgency dimension. These estimates are then calibrated to enhance certainty and support efficient decision-making in the online phase.

Conversely, the third component, i.e., the *online phase* (see Fig. 23), entirely differs from that in the WB-PrPM approach. In this phase, the BB-PrPM approach is operationalized, where the trained ML models provide real-time estimates of intervention factors. These estimates, combined with a resource monitoring and allocation component that represents the capacity dimension, simulate an environment of constrained resources, thereby enriching the learning process for the intervention policy. An online RL algorithm is deployed to learn this policy through trial-and-error within the environment. The RL agent continuously observes the state of ongoing cases, responds in real-time, and adjusts its decisions based on the rewards or penalties received. Over time, the agent iteratively refines its policy, converging toward an optimal intervention policy that maximizes the total gain.

Therefore, this chapter will focus on the BB-PrPM approach, highlighting its distinctions from the WB-PrPM. Where applicable, relevant aspects of the WB-PrPM approach will be referenced to provide context and clarify these differences.

6.2. Factors Affecting Intervention Policies

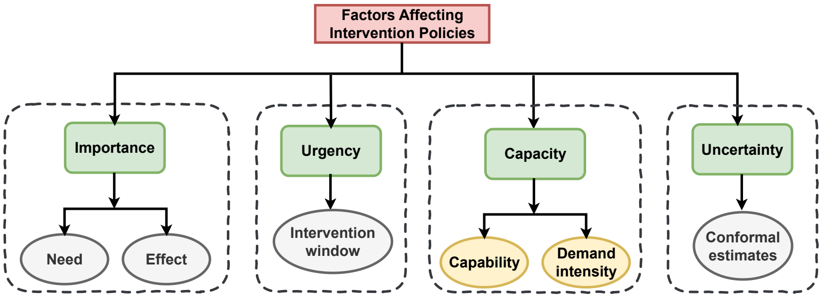


Figure 21. Factors influencing intervention policy decision-making.

The BB-PrPM approach is developed around factors influencing the agent's decision to trigger an intervention and allocate resources to a specific case within a given state. As illustrated in Fig. 21, these factors are categorized into four key dimensions: Importance, Urgency, Capacity, and Uncertainty. Given that the primary difference from the discussion in Section 5.2 pertains to the capacity dimension; this section will focus on the key distinctions within this dimension.

In a resource-constrained environment where resources are limited, the capacity dimension focuses on determining the feasibility of interventions. Even when an intervention is necessary, effective, and urgent, it may become impractical if resources are insufficient or the demand for interventions exceeds available capacity. In such cases, triggering the intervention may not be feasible. To better understand this dimension, we subdivide it into *Capability* and *Demand Intensity*.

Capability refers to the ability and feasibility of executing an intervention based on the availability of sufficient resources. It also reflects the extent to which available resources are utilized. When resources are scarce, the feasibility of an intervention is constrained. To effectively model this, we consider three parameters: a boolean variable indicating resource availability, the actual number of available resources (denoted as n), and the proportion of resources available (represented by η). These parameters are continuously monitored and assessed during process execution.

We incorporate these parameters to estimate the capability to execute interventions, capturing both binary availability and quantitative measures. This holistic approach collectively influences the feasibility of triggering interventions, helping in informed decision-making while resources are limited.

Demand Intensity accounts for situations where sufficient resources exist for an intervention, but the decision to intervene might be postponed. For example, if three resources are available and three cases require intervention, utilizing all resources immediately would consume the entire intervention resource pool, leaving no resources for potential future cases that might yield higher gains from the intervention. To address this issue, we propose incorporating two parameters: the arrival rate (λ) of cases and the work-in-progress (WIP). As the number of incoming cases increases, the demand for available resources for intervention execution also rises, leading to a higher workload.

6.3. Offline phase of the BB-PrPM

The offline phase of the BB-PrPM approach shares similarities with the offline phase of the WB-PrPM approach (see Section 5.3). However, for clarity and completeness, the key aspects are summarized here.

The offline phase has two sub-phases: training and calibration (Fig. 22). In the training sub-phase, event logs are preprocessed and encoded for model training. Various models are trained to address specific intervention policy factors:

- A predictive model: Forecast the need for intervention, parameterized as Outcome Prediction (OP) and Total Uncertainty (TU).
- A causal model: Estimate the effect of the intervention, represented as the intervention effect (IE).
- A survival model: Assess the urgency of the intervention, captured as the Intervention Window (IW).

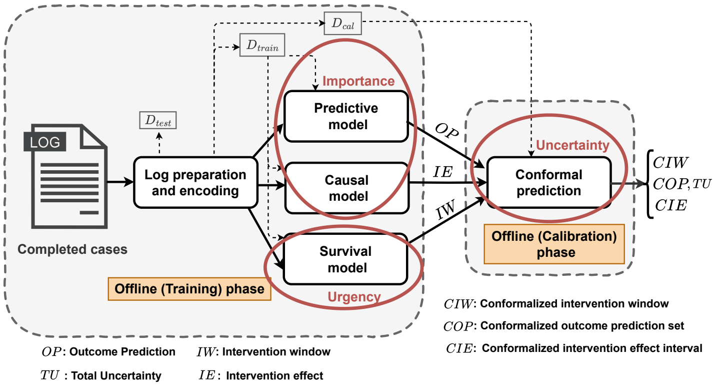


Figure 22. An overview of the offline phase.

The calibration sub-phase calibrates the model predictions to account for uncertainty. Instead of relying on point estimates, predictions are adjusted using conformal prediction techniques (discussed in Section 5.3.4) to produce conformalized estimates. These calibrated predictions are used during the online phase to guide intervention decisions.

Conformal prediction is applied to predictive, survival, and causal estimates (OP, IW, and IE) to ensure they meet a user-defined confidence level, $1-\alpha$. This process transforms the original estimates into Conformalized Outcome Prediction (COP) for OP, Conformalized Intervention Window (CIW) for IW, and Conformalized Intervention Effect (CIE) for IE.

6.4. Online phase of the BB-PrPM

In this phase, depicted in Fig. 23, an online reinforcement learning (RL) algorithm is employed to learn an intervention policy through trial-and-error, thereby operationalizing the BB-PrPM approach. The goal of this phase is for the RL agent to interact within an environment characterized by limited resources, optimizing its policy to maximize cumulative rewards while achieving faster convergence. Over time, the agent incrementally refines its performance by learning from these interactions and iteratively adjusting its policy.

The first step in adapting this scenario into a PrPM task under uncertainty and resource constraints involves utilizing the models trained during the offline phase. These models generate conformal estimates, representing case states at different decision points or prefixes. These estimates capture information about the need (COP and TU), effect (CIE), and intervention window (CIW), reflecting the importance, urgency, and uncertainty of interventions. A capacity monitoring component is then integrated to continuously observe event flow, considering arrival patterns and the availability of intervention resources, as defined by domain-specific knowledge.

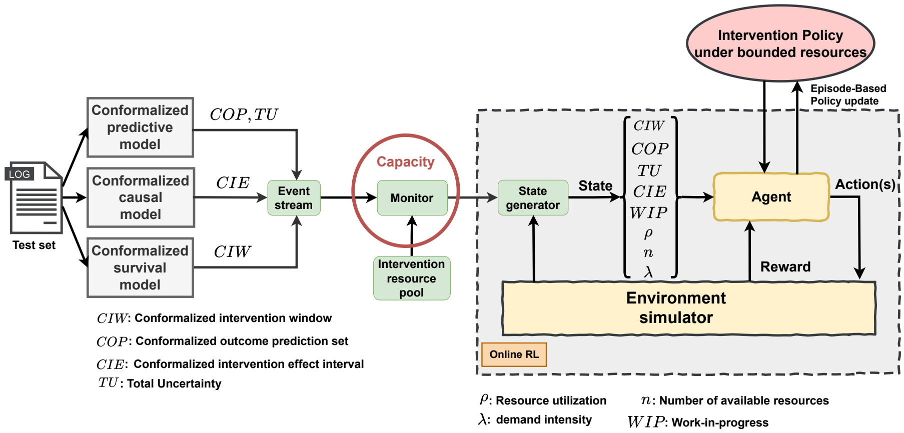


Figure 23. An overview of the online phase.

The capacity monitoring component assumes a continuous stream of new events from ongoing cases during operation or runtime. These events are collated to accumulate the sequence of events, meaning that multiple cases may be ongoing concurrently at any given time step. The agent must then decide which cases require intervention and when, considering the constraints of limited resources. Continuous monitoring of available resources, workload, and demand intensity is necessary. Specifically, the system actively monitors the number (n) and percentage (η) of available resources, the workload represented by cases in progress (WIP), and the intervention demand indicated by the rate of case arrivals (η) during runtime. These metrics are updated whenever a new event arrives or when the agent triggers an intervention, i.e., allocates resources.

For instance, if there are three available resources ($n = 3$ and $\eta = 100\%$) and the first event of the first case arrives, the workload (WIP) is set to 1. If no intervention is triggered, n remains 3, and η remains 100%. At this point, λ is considered zero since only one case exists. However, if another case arrives and the agent decides to trigger an intervention, WIP becomes 2, n becomes 2, and η drops to 66% due to resource allocation. This implies that the allocated resources are temporarily occupied for a specific duration before being released. A resource allocator manages this process, allocating, blocking, and releasing resources during runtime. The arrival rate λ is determined as the inverse of the time difference between the arrivals of the two cases. For example, if the first case arrives at 10:00 AM and the second case arrives at 10:30 AM, the arrival rate would be 2 cases per hour (1/0.5 = 2 cases per hour).

Moreover, the RL agent operates across multiple cases simultaneously to mimic real-life scenarios where cases often arrive concurrently and may overlap. This approach also helps prevent data leakage if information is provided to the agent on a case-by-case or sequential basis. We achieve this by providing the agent with information while simulating a unique case identifier for each incoming event and updating the agent's policy after the completion of any case.

In the following sections, we will explain the learning problem, including constructing the environment with which the agent interacts. Additionally, we will provide a detailed explanation of the reward function.

6.4.1. Learning Problem

The first step in formulating the learning problem is to define the RL agent's environment. This environment is constructed based on states generated by a *state generator*, with each state representing a snapshot of an ongoing case. These states are dynamically updated during process execution, reflecting various parameters at different operational times. The state space includes factors influencing intervention policies, such as the need for intervention (COP and TU), the potential effect of the intervention (CIE), the timing window for intervention (CIW), and factors of workload and resource management (WIP, η, n, and λ). This state representation is meant to enable the RL agent to make informed decisions, accelerate convergence, and optimize the timing and allocation of interventions.

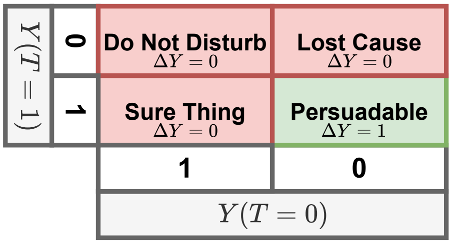
The environment is designed to capture the dynamic changes in process execution, making it both resource-aware and resource-constrained. On one hand, it integrates detailed information about the availability of resources, ensuring that the agent is aware of resource limitations. On the other hand, the environment enforces constraints that prevent the agent from initiating interventions when resources are insufficient, thus simulating a realistic resource-constrained setting.

The agent begins its interaction by observing the initial state S_t, generated by the state generator and corresponding to a specific prefix of an ongoing case. This state provides the agent with the necessary information to make decisions about the case at that particular point in time. The agent must decide whether to intervene immediately, delay, or refrain from intervening altogether.

The action space $A_t \in \{0, 1\}$ at each state or time step presents two options: 0, which denotes the decision not to trigger an intervention, and 1, which indicates the choice to trigger an intervention. Interventions are predefined actions based on domain knowledge that aim to positively influence cases that are likely to end with a negative outcome. For example, an intervention could involve offering a discount to a customer or reducing the interest rate on a loan application to improve its chances of approval.

As the simulation progresses to the next step ($t + 1$), a new state $S_t + 1$ is generated with the arrival of a new event. The agent then receives a reward or penalty (R_t) based on the action A_t taken in the previous state S_t and the resulting consequences. The agent is rewarded if the intervention successfully transforms a negative outcome into a positive one (a "persuadable case"), with the change in outcome denoted as ΔY, as shown in Eq. 6.4.1 and Fig. 24. The ΔY, can be either 0 or 1, where 1 indicates that the outcome changed from negative to positive due to the intervention, and 0 indicates no change. On the other hand, the agent is penalized if the outcome remains negative despite the intervention (a "lost cause" case). Additionally, if the outcome had been positive without intervention (a "sure thing" case), the intervention would have been unnecessary and may have resulted in a penalty. Furthermore, if an initially positive outcome turns negative following an intervention (a "do not disturb" case), this leads to a negative effect that should be avoided.

$$\Delta Y = \begin{cases} 1 & \text{if } Y(T=0) = 0 \text{ and } Y(T=1) = 1 \\ 0 & \text{otherwise} \end{cases}$$



$Y(T=1)$		$Y(T=0)$ = 1	$Y(T=0)$ = 0
	0	**Do Not Disturb** $\Delta Y = 0$	**Lost Cause** $\Delta Y = 0$
	1	**Sure Thing** $\Delta Y = 0$	**Persuadable** $\Delta Y = 1$

Figure 24. Grouping of cases based on intervention outcomes

To assess the effectiveness of interventions, we use the RealCause package [30], which generates alternative potential outcomes for each case prefix: $Y(T=1)$ for the outcome with the intervention and $Y(T=0)$ for the outcome without the intervention, as shown in Fig. 24. This package addresses the challenge of determining intervention effectiveness directly from historical event logs, where only one potential outcome per case is typically observed. The RealCause model estimates what the outcome would have been without the intervention and vice versa, ensuring these alternative outcomes are statistically indistinguishable from the actual observed outcomes.

In this constructed environment, events from the event log are replayed chronologically. The agent makes decisions based on the current state of each case, with unique case identifiers indicating whether a new event pertains to an existing case or marks the start of a new one. This iterative process of observing new states and taking actions based on the specific conditions of each case is fundamental to the agent's learning process. Through exploration and exploitation of various interventions, the agent gradually converges toward an optimal intervention policy that maximizes the total gain. Convergence is achieved when the agent consistently makes decisions that trigger interventions when necessary, effective, and timely, all while considering resource availability. This optimal policy is reflected in continuous positive rewards and positive total gain from the agent's decisions.

6.4.2. Reward Function

The reward function in the BB-PrPM approach is designed to reflect the effectiveness of the agent's intervention decisions. This reward aligns with the primary objective of maximizing the total gain by targeting cases from the persuadable group and increasing the number of cases where the intervention successfully changes the outcome from negative to positive. Also, the reward function must account for minimizing intervention costs and ensuring efficient use of available resources.

To achieve this, the reward function incorporates several key elements: the cost of the intervention (C_{in}), the gain from correctly allocating resources ($Gain_{res}$), the gain from a positive outcome ($Gain$), and the change in outcome (ΔY). The inclusion of $Gain_{res}$ is necessary as it guides the agent to trigger interventions only when the required resources are available. Triggering an intervention without sufficient resources would render it ineffective. Similarly, the positive outcome gain ($Gain$) reflects the benefits of achieving a positive outcome, adjusted for the intervention's cost. This is important because resource-backed interventions can be beneficial but also incur costs that need to be considered. Table 13 provides a detailed illustration of the reward function across different scenarios.

When the agent triggers an intervention with sufficient resources, the reward or penalty it receives depends on the change in outcome, leading to two scenarios:

- **Positive Effect ($\Delta Y = 1$):** The agent earns a reward of $r = (Gain - C_{in}) + Gain_{res}$, reflecting both the positive outcome and the efficient allocation of resources minus the intervention cost.
- **Negative Effect ($\Delta Y = 0$):** The agent incurs a penalty of $r = -(Gain + Gain_{res} + C_{in})$, representing the failure to achieve a positive outcome, resource misallocation, and the cost of triggering an ineffective intervention.

Table 13. The proposed reward function

		Agent triggers the intervention	
Resource available	ΔY	Yes	No
Yes	1	$(Gain - C_{in}) + Gain_{res}$	$-(Gain + Gain_{res})$
	0	$-(Gain + Gain_{res} + C_{in})$	$Gain + Gain_{res}$
No	1	$-(Gain_{res})$	$Gain_{res}$
	0	$-(Gain_{res})$	$Gain_{res}$

If the agent chooses not to trigger an intervention, the reward or penalty depends on the change in outcome:

- **Positive Effect ($\Delta Y = 1$):** The agent incurs a penalty of $r = -(Gain + Gain_{res})$, reflecting the missed opportunity to achieve a positive outcome.
- **Negative Effect ($\Delta Y = 0$):** The agent secures a reward of $r = Gain + Gain_{res}$, avoiding the costs of an ineffective intervention.

In cases where resources are unavailable, it is assumed that interventions cannot be triggered due to these limitations, and the negative outcome cannot be prevented. Here, rewards or penalties are assigned based on the agent's resource allocation decisions:

- If the agent triggers an intervention despite insufficient resources, it suffers a penalty of $r = -Gain_{res}$, reflecting the inefficacy of the action due to resource constraints.
- If the agent refrains from triggering an intervention, it receives a reward of $r = Gain_{res}$, rewarding the cautious decision to avoid intervention when resources are unavailable.

In summary, the reward function integrates the components of the total gain function (positive outcome gain and intervention cost) and emphasizes resource allocation effectiveness. The agent is penalized for misallocating resources to unnecessary interventions and for recommending interventions when no resources are available. Conversely, the agent is rewarded for allocating resources to interventions that successfully change outcomes from negative to positive. This encourages efficient resource management and discourages situations where needed interventions cannot be performed due to resource shortages.

$$TotalGain = (\Delta Y * Gain) - C_{in} \tag{6.1}$$

Finally, the total gain achieved by the RL agent is calculated on an episode basis. An episode begins when the first event of a case is observed and concludes with the last event of that case. The RL agent mitigates the risk of data leakage by updating its policy at the end of each episode, i.e., at the end of each ongoing case. This total gain is computed as outlined in Eq. 6.1, detailed in Section 4.5.2.

6.5. Evaluation

In this section, we evaluate the BB-PrPM approach in terms of its convergence behavior and performance metrics across various levels of resource utilization, addressing RQ3. The evaluation includes a comparative analysis against RL-based baseline approaches [1, 59] that do not consider resource constraints or the inherent uncertainties within the intervention policy. Specifically, the evaluation measures the agent's ability to effectively allocate available resources to execute interventions when they are needed, effective, and urgent.

Convergence Metric: The convergence point is defined as the moment when the RL agent's cumulative gain becomes positive and remains so until the end of the simulation. This metric evaluates the speed—or the number of cases required—for the RL agent to learn an intervention policy that consistently yields positive total gains.

Performance Metric: The performance evaluation focuses on the total gain achieved after reaching the convergence point. This approach is chosen because the online RL agent requires a learning period to determine an optimal policy for achieving positive gains. Assessing performance during this learning duration might not accurately represent the agent's capabilities. Therefore, the total gain prior to convergence can be considered a warm-up period, after which the learned policy can be effectively deployed in a production environment.

To address RQ3 and guide the evaluation process, we divide it into two subquestions, examining the BB-PrPM approach's internal variants and its standing relative to existing baseline methods:

RQ3.1: How do different variants of the BB-PrPM approach perform in terms of convergence (measured by the number of cases) and performance (measured by total gain post-convergence) across various resource utilization levels?

RQ3.2: How does the performance of a particular variant of the BB-PrPM approach compare to baseline methods in terms of convergence and total gain across different resource utilization levels?

6.5.1. Datasets and Experimental Setup

The event logs and experimental setup used to evaluate the BB-PrPM approach are consistent with those described in Sections 5.5.1 and 5.5.2, particularly regarding the offline (training and calibration) phase.

We utilized the same event logs, BPIC2012 and BPIC2017, which represent the loan origination process and are explained in detail in Section 4.2.1. Each log contains detailed information on the sequence of activities performed during the loan application process, including case outcomes and intervention, which are essential for training and evaluating prescriptive approaches.

The experimental setup follows the methodology discussed in Section 5.5.2 and extends it by incorporating adjustments to operate the online RL component. In line with prior research [1, 59], the Proximal Policy Optimization (PPO) algorithm [166] is selected as the online RL algorithm for its ability to optimize policies effectively in continuous control tasks.

The parameters $Gain$ and C_{in} typically require configuration by domain experts to optimize effectiveness, as they vary between business processes. In this experiment, we use a medium-cost-benefit strategy with a $C_{in}/Gain$ ratio of 50%. $Gain_{res}$ is set to 50% of the margin, where the margin represents the potential positive gain outcome minus the intervention cost in a scenario where the agent triggers the intervention and achieves a positive outcome. Specifically, the Gain is set to \$60, representing the potential benefits of achieving a positive outcome. The $Gain_{res}$ is set to \$15, highlighting the importance of efficient resource allocation. The C_{in} is set to \$30, indicating a balanced intervention cost.

These configurations are variable and dependent on the specific process and domain knowledge. Experiments are also conducted with different cost-to-benefit ratios, revealing that higher ratios slow RL agent convergence while lower ratios speed it up. Additionally, experiments where *Gainres* was set to \$0 indicated that providing a signal for efficient resource allocation improved the performance of the RL policy when $Gain_{res}$ was greater than zero.

Furthermore, we followed the same approach discussed in Section 5.5.2 to establish the four levels of resource utilization: high, moderately high (M.High), medium, and low. These levels were determined post-operation for each log, enabling a thorough evaluation of the BB-PrPM approach under varying resource constraints. Accordingly, the experimental setup defines four resource utilization levels: high, moderately high, medium, and low. These levels are defined post-simulation by varying the available resources across different ranges. Through extensive resource range analysis, we identify resource availability thresholds for each utilization level, as shown in Table 12. For instance, in the *BPIC2012* log, resource allocation is set to $n = 1$ for high resource utilization, $n = 4$ for moderately high, $n = 6$ for medium, and $n = 12$ for the low level.

Intervention duration determination varies based on execution time, particularly in one-by-one event streaming experiments. To maintain consistency, we fixed the intervention duration at 1 second across all resource utilization levels. This standardization allows for seamless event streaming and ensures relative consistency across datasets with different case and event counts. However, the primary focus remains on varying resource utilization levels, which fluctuate across datasets within the fixed intervention duration.

Variants of the BB-PrPM Approach. To evaluate the effectiveness of the BB-PrPM approach across different levels of resource utilization, we introduce four variants: "*All*," "*withCATE*," "*withoutCIW*," and "*withoutTU*." Each variant generates a distinct intervention policy, affecting both convergence and overall performance, primarily by altering the state representations provided to the RL agent.

- *All Variant:* This variant reflects the full BB-PrPM approach as introduced. It incorporates all the factors discussed that could influence intervention policies. The state representation in this variant includes parameters related to need (COP and TU), effect (CIE), intervention window (CIW), and workload and resource management (WIP, η, n, and λ).
- *withCATE Variant:* In this variant, the CIE is replaced by the lower and upper bounds of the $CATE$, as derived from prior research [1]. The state representation includes: CIW, COP, TU, $CATE_{lb}$, $CATE_{ub}$, WIP, η, n, and λ. This variant allows for a direct comparison between the use of individual intervention effects, represented by the CIE, and conditional average treatment effects, represented by $CATE_{lb}$ and $CATE_{ub}$, in guiding the RL agent towards intervention policies. It helps determine which intervention effect measure leads to quicker convergence and better overall performance.
- *withoutCIW Variant:* In this variant, the RL agent lacks explicit information about the intervention window (CIW), which represents remaining intervention time and urgency. The state representation includes: COP, TU, CIE, η, n, and λ. This variant tests whether the RL agent can autonomously infer optimal intervention timing without CIW information, assessing the importance of urgency-related data.

- *withoutTU Variant:* In this variant, the RL agent is not provided with information regarding the Total Uncertainty (TU) linked to the outcome prediction. The state representation includes: CIW, COP, CIE, WIP, η, n, and λ. By withholding TU information, this variant evaluates whether supplying quantified uncertainty information enhances the RL agent's ability to prioritize cases, allocate resources more effectively, and converge quickly towards policies that maximize total gain.

Each of these variants allows for a detailed examination of how specific factors within the BB-PrPM approach influence the overall effectiveness of intervention policies, particularly under varying resource utilization levels. This comparative analysis helps in understanding the role of different factors in optimizing the performance of the BB-PrPM approach.

6.5.2. Results

In this section, we present the results of evaluating the learned intervention policy across various resource utilization levels, focusing on two key measures: *convergence*, which denotes the point at which the RL agent consistently makes decisions resulting in positive gains, and *performance*, which assesses the total gain achieved after convergence.

Comparative Analysis of BB-PrPM Approach Variants. To address RQ4.1, we investigate how the RL agent's convergence behavior varies with different levels of resource utilization. As the availability of resources increases, the agent may take longer to learn the effects of resource saturation due to the greater complexity of managing more resources for interventions. This could lead to a longer time to achieve convergence.

Table 14. Convergence speed, measured in terms of the number of cases, for different proposal variants across various resource utilization levels and logs.

		Resource utilization			
Log	Variant	High	Moderately High	Medium	Low
BPIC2012	All	**629**	**1024**	1158	-
	withCATE	884	1132	**1136**	1142
	withoutCIW	892	1159	-	-
	withoutTU	772	1155	1138	**769**
BPIC2017	All	**7616**	**222**	6572	-
	withCATE	-	7595	7610	5072
	withoutCIW	-	3687	7594	**1707**
	withoutTU	7735	4933	**3723**	-

Table 14 presents the convergence rates for different variants of the BB-PrPM approach across various resource utilization levels using the *BPIC2012* and *BPIC2017* logs. The "*All*" variant generally converges the fastest under high and moderately high resource utilization levels in both logs. This suggests that when resources are limited, explicitly incorporating all factors affecting intervention policies can help the agent converge more quickly.

In contrast, in the *BPIC2017* log, the "*withoutCIW*" variant performs well under low resource utilization because this log is large enough to contain sufficient business cases. Thus, when the number of cases or their duration is substantial, the RL agent can infer the optimal timing for interventions without needing explicit information about the intervention window (*CIW*). Additionally, the "*withoutTU*" variant converges faster under medium resource utilization than others. This might be because, with abundant resources, total uncertainty (TU) is unnecessary, as a strict or conservative policy is unneeded.

Similarly, in the *BPIC2012* log, the "*withoutTU*" variant also converges more quickly than the others under low and medium resource utilization levels. This further supports the notion that when resources are plentiful, excluding TU can simplify decision-making, leading to faster convergence. Moreover, the withoutTU variant performs comparably to the "*withCATE*" variant under medium resource utilization, suggesting that simplifying the model may not significantly impact performance in some cases.

In summary, the "*All*" variant demonstrates strong performance, particularly under high and moderately high resource utilization levels. This suggests that incorporating all relevant factors into the state representation may offer a balanced approach to resource management, especially when resources are constrained. However, in scenarios where resource utilization is low or medium, simplifying the model by excluding certain factors, such as TU or *CIW*, can lead to faster convergence, potentially indicating more efficient use of resources.

Table 15. Total gain (in Thousands) post-convergence for different proposal variants across various resource utilization levels and logs.

		Resource utilization			
Log	Variant	High	Moderately High	Medium	Low
BPIC2012	all	**6.8**	**1.6**	0.1	0
	withCATE	3.8	0.5	0.3	0.4
	withoutCIW	4.3	0.1	0	0
	withoutTU	5	0.1	**0.4**	**4.5**
BPIC2017	all	**1.1**	**103.2**	14.3	0
	withCATE	0	0.9	0.8	45.8
	withoutCIW	0	48.1	1.4	**78.5**
	withoutTU	0.1	47.8	**42.2**	0

On the other hand, Table 15 presents the results concerning the performance measure, specifically the total gain achieved post-convergence across the different variants of the BB-PrPM approach under varying levels of resource utilization. The findings show that under conditions of limited resources—where resource utilization is high or moderately high—the "*All*" variant generally outperforms the other variants. This suggests that when resources are constrained, incorporating all relevant factors into the intervention policy allows the RL agent to make more informed decisions, allowing for optimal resource allocation and results in higher total gains when resources are scarce.

In contrast, simplifying the model by excluding parameters like TU or *CIW* can improve performance when resources are abundant, particularly under medium or low resource utilization. This likely occurs because, with abundant resources, a more relaxed intervention policy can be used. As a result, a simpler model converges faster and more efficiently by focusing on key factors. For example, the "*withoutTU*" variant performs well under medium resource utilization, suggesting that excluding total uncertainty enables the model to prioritize primary factors, leading to better resource allocation and higher total gains.

The differences in convergence and performance between the *BPIC2012* and *BPIC2017* logs highlight the impact of log characteristics, such as size and case diversity, on model effectiveness. In larger and more diverse logs like *BPIC2017*, the model may benefit from simplifying assumptions (e.g., excluding *CIW*), as the richness of the data allows the RL agent to infer optimal policies without needing explicit information for every factor. Conversely, in smaller logs like *BPIC2012*, the "*All*" variant, which contains all factors affecting intervention policies, may compensate for the small or lower richness of the data.

Evaluating the Best-Performing Variant Against Baseline Approaches. To address RQ4.2, we conducted experiments comparing a specific variant of the BB-PrPM approach, referred to as the "*All*" variant, against two baseline methods introduced in previous studies [1, 59]. The "*All*" variant was selected for these comparisons because it demonstrated faster convergence and higher total gain, particularly under high or moderately high resource utilization conditions, reflecting scenarios where resources are limited. The two baseline methods used for comparison are referred to as *BL1* (proposed in [1]) and *BL2* (proposed in [59]). Both baselines assume the availability of sufficient resources to execute immediate interventions for all cases and do not account for any resource constraints.

The distinction between the "*All*" variant and the baselines lies in the state representation. The state representation for the "*All*" variant includes parameters such as *COP*, TU, *CIE*, *CIW*, *WIP*, η, n, and λ, encompassing all factors affecting intervention policy. In contrast, the state representation for *BL1* includes only $CATE_{lb}$, $CATE_{ub}$, and *COP*, which focus solely on the importance dimension of intervention factors. The *BL2* state representation includes a relative position, updated after the arrival of a new event to reflect the current prefix relative to the case length, a reliability estimate indicating the agreement or disagreement among classifiers in an ensemble, and the outcome prediction.

It is important to note that *BL1* and *BL2* were originally designed without considering resource constraints. To adapt these baseline methods to a resource-constrained setting, we modified their reward structure by assigning a reward of zero to the agent when it triggers an intervention in a state where no resources are available. This allows for a fair comparison with the "*All*" variant, which is both resource-aware and resource-constrained. The baseline methods, however, remain resource-unaware—they do not incorporate resource information as part of their input—but are executed within a resource-constrained environment.

BPIC2012. In the analysis of the BPIC2012 log, the "*All*" variant of the BB-PrPM approach consistently outperforms the baseline methods in terms of both convergence and total gain across various levels of resource utilization, as illustrated in Fig. 25. Particularly under high and moderately high resource utilization levels, the BB-PrPM approach converges toward an intervention policy that maximizes total gain, demonstrating superior performance compared to the baseline methods. This suggests that the BB-PrPM approach is more effective in resource-constrained environments where managing limited resources is necessary.

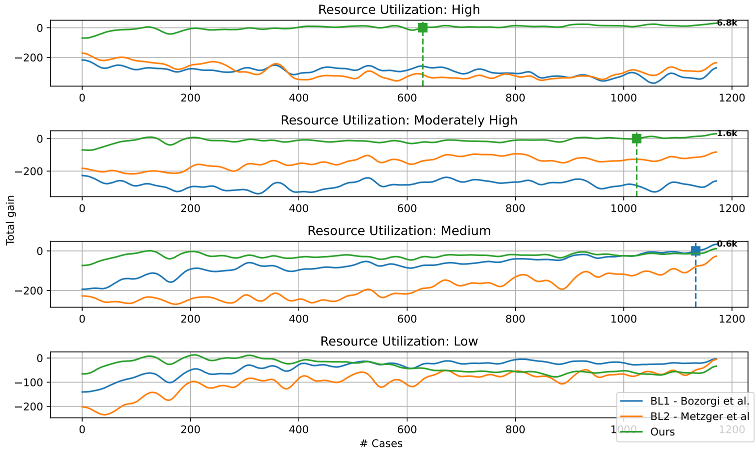


Figure 25. Comparative analysis of the "*All*" Variant and two baseline methods: *BPIC2012*

Conversely, under medium or low levels of resource utilization, the analysis reveals that none of the methods, except for *BL1*, achieve convergence or generate a positive gain. This indicates potential difficulties in reaching convergence in scenarios with more abundant resources, leading to a decrease in overall gain for all methods except *BL1*. The ability of *BL1* to achieve convergence and positive gain under these conditions might be attributed to its simpler state representation, which could be better suited for environments where resources are more readily available. The simplicity of *BL1*'s state representation may enable more straightforward decision-making, particularly when resources are less constrained.

Additionally, it is noteworthy that *BL2* outperforms *BL1* at moderately high resource utilization levels. This improved performance of *BL2* could be linked to its state representation, which includes information reflecting the urgency of interventions, such as the relative position. The inclusion of this information may provide *BL2* with an advantage in scenarios where timely decision-making is necessary, enhancing its ability to allocate resources effectively.

BPIC2017. The analysis of the BPIC2017 log reveals that the "*All*" variant of the BB-PrPM approach consistently outperforms both baseline methods (*BL1* and *BL2*) across all levels of resource utilization, as illustrated in Fig. 26. This superior performance could be attributed to the specific characteristics of the BPIC2017 log, which differs from the BPIC2012 log. Notably, the BPIC2017 log includes cases with a longer average duration, meaning that the RL agent encounters more events per case. This extended exposure enables the agent to learn better and converge toward intervention policies that yield a positive total gain. However, this trend is less apparent when resource utilization is low, likely due to the abundance of available resources that diminishes the challenge of resource constraints.

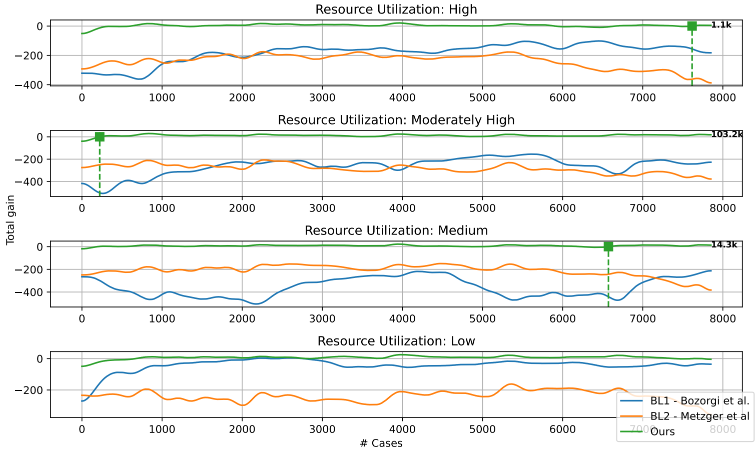


Figure 26. Comparative analysis of the "*All*" variants and two baselines: *BPIC2017*.

Moreover, the "*All*" variant of the BB-PrPM approach exhibits faster convergence toward the optimal intervention policy under moderately high resource utilization. This could be due to the more relaxed resource constraints in such scenarios, which provide the agent greater decision-making flexibility. Additionally, the method of defining the convergence point may influence these results. In this analysis, we reset the total gain to zero if the agent converges and then experiences regression. This ensures that only stable and consistent positive-gain policies are considered as converged. However, it also suggests that the RL agent may not achieve a stable, positive-gain policy within the tested time frame for other resource utilization levels.

When comparing the two baseline methods across different resource utilization levels, it is observed that *BL1* performs better according to the BPIC2017 log than in the BPIC2012 log. This improvement can be attributed to the fact that *BL1*, with its simpler state representation, can still guide the agent in determining an effective intervention policy. However, this simplicity may also lead to longer convergence times before achieving positive gains. This observation highlights the importance of providing the agent with explicit information on factors influencing intervention decisions, especially in resource-constrained environments and during runtime.

The findings from RQ4.1 and RQ4.2, as discussed above, indicate that explicitly providing the RL agent with information regarding the importance, urgency, uncertainty, and capacity dimensions generally results in a more effective intervention policy. The "*All*" variant, in particular, consistently performs well across different resource utilization levels compared to the baseline methods. This trend is especially notable when resources are limited and intervention decisions are subject to uncertainty. In such resource-constrained and uncertain environments, the advantages of equipping the RL agent with information to guide decision-making become clear. This is particularly important in business process management, where operational settings vary significantly, necessitating tailored intervention policies, especially during real-time operations.

6.5.3. Threats to Validity

In addition to the external threats to validity discussed in Section 4.5.4, the evaluation of the BB-PrPM approach presents specific threats related to internal and ecological validity.

Internal Validity: The stochastic nature of the reinforcement learning agent's learning process introduces potential threats to internal validity. To mitigate this stochasticity, we conducted each experiment three times and reported the average results across these replications. Although minor variations in convergence rates and total gains were observed, the results remained consistent across different runs, reinforcing the reliability of our findings.

Ecological Validity: Potential threats to ecological validity stem from the assumption that all resources are equally proficient in executing interventions. Moreover, the BB-PrPM approach assumes a single type of intervention, whereas, in real-world scenarios, there may be multiple types of interventions, such as offering a discount or providing a personalized consultation to a customer. The proposed method does not account for such "multi-intervention" settings, which may limit its applicability in more complex, real-world environments.

6.6. Summary

This chapter introduced the Black-Box Prescriptive Process Monitoring (BB-PrPM) approach, designed to enhance business process performance by triggering runtime interventions. Unlike the white-box approach discussed earlier, the BB-PrPM approach uses online reinforcement learning to optimize intervention policies in environments with resource constraints and uncertainty. By abstracting the complexity of policy formulation into a data-driven model, the BB-PrPM approach reduces reliance on manual input from business analysts, who traditionally need to specify key parameters and determine the conditions and timing for interventions.

To achieve this, the BB-PrPM approach integrates predictive and causal models to estimate interventions' necessity and expected effects, uses a survival model to assess urgency, and applies conformal prediction techniques to manage uncertainty. It also incorporates a resource capacity monitoring component, ensuring that resource allocation is an integral part of decision-making. Through online reinforcement learning, the approach dynamically learns which cases require intervention and when within the context of uncertain and resource-limited environments. Consequently, the resulting intervention policy is both resource-aware and resource-constrained.

We conducted a comparative analysis of various BB-PrPM approach variants and benchmarked the best-performing variant against two baseline methods from the state-of-the-art. The evaluation revealed that the variant incorporating information on the importance, urgency, uncertainty, and capacity dimensions generally outperforms others, achieving faster convergence and higher total gains. This was particularly evident under high or moderately high resource utilization levels, reflecting scenarios with limited resources.

In the next chapter, we conclude this thesis and highlight promising directions for future work in the area of prescriptive analytics for business processes.

7. CONCLUSION

7.1. Summary of Contributions

This thesis investigated two shortcomings of existing approaches to prescriptive process monitoring (PrPM). First, existing PrPM approaches do not account for the uncertainty of the machine learning models they rely upon. This limitation may lead to interventions being triggered in situations where the predictions and other estimates produced by these models are unreliable, leading to suboptimal outcomes. Second, existing prescriptive monitoring approaches assume that it is possible to trigger an unlimited number of interventions during any period of time. In many use cases, however, each intervention blocks a resource (e.g., a process worker) during a certain period of time. Assuming that the number of resources is limited, this observation entails that a prescriptive process monitoring system can only trigger a limited number of interventions during a given period of time.

The thesis addressed the first gap by extending existing prescriptive process monitoring methods with mechanisms to account for prediction uncertainty. Specifically, we experimented with two mechanisms for uncertainty modeling: (i) uncertainty scores derived from ensemble methods; and (ii) conformal predictions, which provide formal guarantees on the confidence of predictions. We found that the latter approach is particularly promising in the context of PrPM.

The thesis then addressed the second gap by proposing both white-box and black-box PrPM methods that explicitly account for resource capacity. The main hypothesis underpinning the proposed resource-aware PrPM methods is that, when resources are limited, the intervention policy needs to take into account the following factors:

- The necessity of triggering an intervention, which is be modeled using a classification model that predicts whether or not a case will end up in a negative outcome.
- The effectiveness of an intervention, which is modeled using a causal effect estimation model.
- The urgency of the intervention (can the intervention be postponed?), which is modeled by means of a survival model.
- The resource capacity available to perform the intervention.
- And the uncertainty of the previous models (e.g. uncertainty of the classification models), which is captured using conformal prediction methods.

7.2. Research Questions Revisited

Research Question 1: *How can we develop intervention policies that effectively incorporate uncertainty into decision-making, ensuring that predictions are reliable and decisions are made confidently to maximize a total gain function?*

Chapter 4 focused on integrating uncertainty into a predictive monitoring approach to create an uncertainty-aware intervention policy. The goal is to avoid triggering interventions when the level of uncertainty is high.

Specifically, we introduced an uncertainty-aware prescriptive process monitoring approach, namely UN-PrPM, which enhances traditional predictive monitoring approaches by incorporating two mechanisms to account for uncertainty (uncertainty scores and conformal predictions), thus enabling more reliable intervention policies. At runtime, both the predictions and their associated uncertainty measures are utilized to operationalize the UN-PrPM approach. This allows the PrPM system to determine which cases necessitate intervention while also considering that interventions, though potentially beneficial, carry associated costs.

An empirical evaluation of the UN-PrPM approach, conducted using real-world event logs, demonstrated its effectiveness in enhancing process outcomes compared to baseline methods that do not factor in uncertainty. The evaluation showed that uncertainty-aware policies are generally more conservative, achieving a comparable total gain with fewer interventions but higher success rates.

Research Question 2: *How can business stakeholders define transparent, rule-based white-box intervention policies that manage limited resources efficiently while optimizing the total gain of the resulting intervention policy in a resource-constrained setting?*

Chapter 5 focused on developing a white-box PrPM approach (namely WB-PrPM) that enables business stakeholders to specify interpretable, rule-based methods for triggering interventions on the basis of multiple relevant factors. This approach is designed to be both uncertainty-aware and resource-constrained, making it particularly effective when decision-makers must balance the demand for interventions with the limitations of available resources.

The WB-PrPM approach identifies key factors influencing intervention policies under resource constraints, categorized into four top-level dimensions: importance, urgency, uncertainty, and capacity. These factors were estimated using relevant machine learning techniques, such as predictive, causal, and survival models. These estimates were then integrated with a filtering and ranking policy to identify cases requiring intervention and select the most appropriate one. Additionally, a resource allocator was incorporated into the approach to manage allocating, blocking, and releasing resources from the intervention resource pool.

The evaluation of the WB-PrPM approach highlighted its effectiveness in balancing resource utilization and process optimization. The assessment studied the performance of distinct variants, mainly in terms of total gain at various resource utilization levels. It also compared WB-PrPM with baseline methods that focused on intervention policy but lacked consideration of resource constraints and inherent uncertainty. By factoring in importance, urgency, and uncertainty, resources are allocated where most needed, maximizing total gain from interventions.

Research Question 3: *How can we design data-driven, black-box intervention policies that manage limited resources effectively, optimizing resource allocation and process performance, i.e., total gain, within a resource-constrained setting?*

Chapter 6 focused on developing data-driven, black-box intervention policies that optimize resource allocation and process performance without requiring manual policy specification, as required in the WB-PrPM approach. While WB-PrPM offers transparency and control, it also demands significant input from process workers, who must thoroughly understand process dynamics. This manual approach can be time-consuming and prone to errors, especially in complex scenarios where multiple factors interact unpredictably. To address these challenges, we introduced the black-box prescriptive process monitoring approach (BB-PrPM), which utilizes online reinforcement learning (RL) to develop intervention policies at runtime. The BB-PrPM abstracts the complexity of policy formulation into a data-driven model, allowing it to adapt to changing process conditions and resource availability without needing constant human input.

The BB-PrPM approach builds upon the WB-PrPM approach, extending it into a data-driven paradigm. Similar to the WB-PrPM, the BB-PrPM approach leverages predictive and causal models to estimate the necessity and expected impact of interventions. It also incorporates survival models to evaluate the urgency of interventions and employs conformal prediction techniques to address uncertainty. Additionally, the BB-PrPM includes a capacity-monitoring component that tracks resource allocation, utilization, and demand intensity, ensuring these factors are integrated into the intervention policy. The primary distinction between WB-PrPM and BB-PrPM lies in the black-box nature of the latter's intervention policies, where the decision-making process is driven by data and automated learning rather than predefined rules.

Empirical results demonstrate that the BB-PrPM optimizes process performance while handling resource constraints. By continuously learning from ongoing process executions, the BB-PrPM can dynamically adjust intervention policies to achieve optimal outcomes, reducing the need for manual intervention and minimizing the risk of human error. The empirical evaluation assessed the effectiveness of various BB-PrPM approach variants, focusing on convergence and performance metrics across different levels of resource utilization. This evaluation compared the BB-PrPM to baseline RL-based methods that do not account for the policy's resource constraints and uncertainty measures. The results highlight the BB-PrPM's ability to rapidly converge toward intervention policies that maximize total gain post-convergence, especially when resource utilization is high or moderately high, illustrating its potential for enhancing process performance in resource-constrained environments.

7.3. Comparison of Proposed PrPM Approaches

Table 16 compares the three proposed PrPM approaches (UN-PrPM, WB-PrPM, and BB-PrPM) across several aspects, including their goals, resource constraints, intervention factors, decision procedure, methodology, adaptability, use cases, and performance evaluation.

Table 16. Comparison of the three presented Prescriptive Process Monitoring (PrPM) Approaches: UN-PrPM, WB-PrPM, and BB-PrPM

Aspect	UN-PrPM	WB-PrPM	BB-PrPM
Goal	Uncertainty-aware intervention policies to maximize total gain.	Uncertainty-aware, resource-constrained, and rule-based intervention policies.	Uncertainty-aware, resource-constrained, data-driven intervention policies to maximize total gain.
Resource Constraints	Unlimited resources.	Limited resources.	Limited resources.
Intervention Factors	Intervention need and uncertainty.	Intervention need, effectiveness, urgency, capacity, uncertainty.	Intervention need, effectiveness, urgency, capacity, uncertainty.
Decision Procedure	Filters and prioritizes cases based on uncertainty-aware predictions.	Users define thresholds and weights; employs filtering and ranking for prioritization (Priority Scheduling).	Autonomously learns intervention policies from historical data.
Methodology	Triggers interventions based on predictive scores and uncertainty measures, according to empirically determined thresholds.	Combines predictive models, causal inference, survival analysis, and rule-based mechanisms with resource allocation.	Uses RL to learn intervention policies from historical data.
Adaptability	Enhances prediction reliability without considering operational constraints.	Balances conservative and aggressive policies depending on resource availability, using filtering and ranking.	Learns to allocate resources to maximize its rewards function under different scenarios.
Use Cases	Suitable for environments where resource constraints are not critical.	Best for scenarios requiring explainable policies and effective resource management.	Ideal for complex, resource-constrained environments with sufficient historical data for RL training.
Evaluation Results	Uncertainty-aware policies are more conservative and achieve higher success rates than non-uncertainty-aware baselines, but, overall, they achieve comparable total gain.	Consistently achieves higher total gains than baselines by incorporating causal effects, urgency, and prioritizing time-sensitive cases. Simpler uncertainty-aware variants may outperform more complex ones.	RL policies that incorporate importance, urgency, uncertainty, and capacity (the “complex variant”) lead to faster convergence and better performance, especially in resource-constrained settings.
Performance of Simple vs. Complex Variants	-	Simple variants (e.g., *IP*10) achieve higher Total Gain compared to complex variants (e.g., *IP*15).	Complex variants (e.g., the *All* variant) outperform simple ones, especially under high resource utilization.
Performance with High vs. Low Resource Utilization	-	Performs better, under low resource utilization (e.g., *IP*10 in BPIC2012).	Excels under high resource utilization (e.g., "*All*" variant in BPIC2017).
Best Performing Variant	-	- BPIC2012: Urgency + conformal policy (*IP*10) across all resource utilization levels. - BPIC2017: Causal policy (*IP*2/*IP*8) adapts to resource shifts (IP2 for High/M. High, *IP*8 for Medium/Low).	- Dominated by the "*All*" variant under High/Moderately High resource utilization levels. - "*WithoutTU*" variant performs best under Medium/Low resource utilization levels.

The UN-PrPM approach maximizes total gain with uncertainty-aware intervention policies, assuming unlimited resources. In contrast, the WB-PrPM approach integrates resource constraints via rule-based allocation mechanisms, balancing factors like importance, urgency, uncertainty, and capacity. This makes it suitable for scenarios needing explicit policies and resource management. However, while adaptable, it may not perform as well in highly dynamic environments compared to data-driven approaches.

The BB-PrPM approach, which uses reinforcement learning (RL), models both uncertainty and resource constraints. It is highly adaptive and suitable for complex, resource-constrained environments, especially when there is enough historical data for training. While it requires more data and computational resources, it consistently outperforms the other two approaches in resource-constrained scenarios. Each of the three approaches offers distinct advantages, with UN-PrPM being the simplest, WB-PrPM providing transparency, and BB-PrPM being the most dynamic and adaptable.

Additionally, the comparison in Table 16 reveals distinct performance patterns between the white-box (WB-PrPM) and black-box (BB-PrPM) approaches. For WB-PrPM, simpler, interpretable variants that contain fewer than or equal to two parameters (e.g., IP10 in BPIC2012) consistently outperform more complex variants with more than two parameters (e.g., IP15) across various resource utilization levels. In contrast, BB-PrPM relies on complex variants, such as the "All" policy, which dominates in high resource utilization scenarios but underperforms in low resource utilization scenarios. Furthermore, the BB-PrPM approach performs better under high resource utilization levels, while the WB-PrPM approach improves under low resource utilization levels.

Finally, to explicitly compare the performance of the black-box and white-box approaches, we present the results in Table 17. This table compares the WB-PrPM and BB-PrPM approaches based on their performance. It specifically focuses on the total gain (in thousands) achieved by the best-performing variants across various resource utilization levels for two datasets: BPIC2012 and BPIC2017. For each dataset, the table shows the best variant for each approach under four resource utilization categories: High, Moderately High (M. High), Medium, and Low.

Table 17. Comparison of WB-PrPM and BB-PrPM Approaches Based on Total Gain of Best Variant at Different Resource Utilization Levels

		Resource Utilization							
		High		M. High		Medium		Low	
Dataset	Approach	Best Variant	$TotalGain$	Best Variant	$TotalGain$	Best Variant	$TotalGain$	Best Variant	TotalGain
BPIC2012	WB-PrPM	IP10	0.184	IP10	0.349	IP10	0.505	IP10	0.468
	BB-PrPM	All	6.8	All	1.6	$WithoutTU$	0.4	$WithoutTU$	4.5
BPIC2017	WP-PrPM	IP2	0.237	IP2	0.436	IP8	0.462	IP5	0.453
	BB-PrPM	All	1.1	All	103.2	$WithoutTU$	42.2	$WithoutCIW$	78.5

In BPIC2012, the WB-PrPM approach consistently performs well with the IP10 variant (urgency + conformal) across all resource utilization levels, achieving the highest total gain in each category. On the other hand, BB-PrPM's "All" variant performs best regarding high resource utilization. However, it significantly underperforms in other categories, with the "$WithoutTU$" variant achieving better results in medium and low resource utilization.

In BPIC2017, the WB-PrPM approach performs well with the IP2 variant (causal only) at high and moderately high resource utilization, while the IP8 variant (causal + conformal) excels at medium utilization, and IP5 (urgency only) shows the best performance at low utilization. In comparison, the "All" variant of the BB-PrPM approach again performs better at high resource utilization but performs inconsistently across other levels, with the "*WithoutTU*" and "*WithoutCIW*" variants yielding varying results.

This comparison highlights that, in general, and particularly when resources are limited (i.e., high resource utilization), the BB-PrPM approach achieves higher total gains compared to the WB-PrPM approach. It also emphasizes that the performance of the WB-PrPM approach tends to be more stable across different resource utilization levels, though with lower total gains compared to BB-PrPM. In contrast, BB-PrPM's performance is more variable, excelling primarily under high resource utilization, where resources are low. However, it struggles under lower resource utilization due to insufficient data for the RL agent to learn an optimal intervention policy when resources are large.

7.4. Future Directions

While this thesis has addressed key challenges regarding uncertainty and resource constraints in prescriptive process monitoring, several avenues for future research remain. In the following, we outline five possible directions for advancing the field of prescriptive process monitoring in business processes.

7.4.1. Expanding the Dataset Pool

The research presented in this thesis was limited by the selection of available event logs containing case outcomes and interventions necessary for establishing causal relationships. Consequently, the evaluation was restricted to datasets with documentation referencing a case outcome and an intervention. Future research should focus on expanding the range of evaluation datasets to enhance the generalizability of prescriptive process monitoring approaches. This expansion could involve identifying new event logs that meet the necessary criteria or creating synthetic datasets that accurately reflect real-world scenarios.

Moreover, causal discovery techniques could automatically discover causal relationships between interventions and outcomes. These techniques can identify potential interventions within observational data and their causal impact on identified outcomes. However, a limitation of this approach is that the discovered interventions may require further validation by domain experts. In this thesis, we chose to rely on domain experts to directly determine possible interventions, avoiding the additional complexity of discovering and then verifying them.

Expanding the dataset pool will significantly improve the generalizability of prescriptive process monitoring approaches, allowing for more comprehensive evaluations across various business domains.

7.4.2. Evaluation Through Randomized Trials

One of the significant challenges in any method relying on causal inference, such as prescriptive process monitoring, is accurate evaluation. In the research presented in this thesis, historical process execution, i.e., event logs, were used to evaluate prescriptive process monitoring. However, in such observation data cases where an intervention is recorded, we only know the outcome with the intervention, and we lack information about the outcome if the intervention had not occurred. Conversely, for cases where no intervention is recorded, we only know the outcome without the intervention. To estimate these alternative outcomes to identify persuadable cases, we employed the RealCause method, which, while theoretically well-founded and extensively evaluated, may still produce incorrect estimates in some instances.

Given these limitations, a promising direction for future research is using randomized controlled trials (RCTs), which are considered the gold standard for evaluating causal inference-based methods. RCTs involve randomly assigning cases to either receive an intervention or not, allowing for a more accurate assessment of the intervention's true impact by directly comparing outcomes between treated and untreated cases. Integrating RCTs into the evaluation of prescriptive process monitoring approaches would provide stronger validation of their effectiveness. It could greatly enhance the credibility and accuracy of causal inferences drawn from these methods.

7.4.3. Multi-Agent and Hierarchical Resource Allocation

The current resource-constrained approaches in prescriptive process monitoring assume that all resources are homogeneous and equally proficient in executing interventions. This simplification may not hold in more complex, real-world settings where resources often have varying levels of expertise, and decision-making processes may involve multiple layers of hierarchy. A possible future research direction could explore the integration of multi-agent systems and hierarchical resource allocation models.

In a multi-agent system, different agents—such as employees, teams, or departments—possess varying capabilities and responsibilities. These agents could be modeled with distinct proficiencies in executing specific types of interventions. Hierarchical models could then be employed to represent the layered decision-making structures often found in organizations, where decisions are made at different levels depending on the nature of the task and the expertise required.

Such an approach would allow for a more accurate representation of real-world operational environments. By accounting for the diversity in resource capabilities and the complexity of decision-making hierarchies, these models could lead to the development of more effective and tailored intervention policies, enhancing the overall performance of prescriptive process monitoring systems.

7.4.4. Multi-Objective Intervention Policies

In many business processes, decision-makers navigate trade-offs between competing objectives such as cost efficiency, customer satisfaction, and regulatory compliance. Current prescriptive process monitoring approaches typically focus on optimizing a single objective, e.g., improving process outcomes or reducing cycle time. This may limit their applicability in real-world scenarios where multiple goals must be balanced simultaneously.

Future research could address this limitation by developing multi-objective intervention policies that account for these trade-offs. Incorporating methods from multi-objective optimization and reinforcement learning, researchers could design systems that balance different objectives, leading to more refined intervention strategies that better align with several organizational goals.

A promising research direction involves tackling the challenges of a multi-objective reinforcement learning task. Unlike the current work, which focuses on enhancing a single objective (such as reducing negative case outcomes like customer rejections), the problem can be reframed to address multiple objectives concurrently. For example, a system could aim to reduce negative outcomes while also minimizing cycle time or operational costs. By exploring multi-objective RL strategies, future research could provide a more general approach to process optimization and decision-making in complex scenarios where competing objectives are at play.

7.4.5. Multi-Intervention Settings

The prescriptive process monitoring approaches presented in this thesis operate under the assumption that only one type of intervention can be applied (e.g., sending an additional loan offer) and that an intervention is triggered at most once per case. However, real-world business processes often have multiple possible intervention types (such as reducing the monthly interest rate, extending the loan duration, or offering personalized consultations). Moreover, multiple interventions of the same or different types may be triggered within a single case, either simultaneously or sequentially.

Extending the current approaches to accommodate multi-intervention scenarios represents a significant avenue for future research. This would involve designing policies considering the interactions between different interventions and their combined effects on process outcomes. For instance, offering a discount in conjunction with a personalized consultation might be more effective than either intervention alone.

Therefore, future research would investigate the development of prescriptive monitoring systems capable of handling these complex, multi-intervention settings. This would require integrating more sophisticated decision-making algorithms that can evaluate the cumulative impact of various interventions over time, such as Multi-Armed Bandits, thereby improving the flexibility and effectiveness of intervention strategies in real-world applications. By addressing the complexities of multi-intervention scenarios, these advanced systems could better support business processes that require dynamic and adaptive strategies to optimize process performance across various dimensions, such as customer satisfaction, cost efficiency, and compliance.

7.4.6. Handling Multi-Class Outcomes

As process monitoring evolves, prescriptive tasks involving multi-class outcomes, such as customer satisfaction levels $(1-5)$, present challenges beyond binary classification. Key issues include estimating intervention effects across all categories, selecting appropriate evaluation metrics, and addressing class imbalance, where underrepresented levels (e.g., satisfaction levels 1 and 5) can lead to biased predictions.

Estimating intervention effects across all outcomes is crucial. This can be achieved using hierarchical causal models [336] that decompose multi-class outcomes into nested binary decisions (e.g., satisfaction ≥ 3 vs. < 3) or through multi-task learning frameworks that jointly model interventions across all categories [337]. To mitigate class imbalance, techniques such as SMOTE [338], cost-sensitive learning [339], and ensemble methods like XGBoost can be employed. Additionally, evaluation metrics like the weighted F1-score, MCC, and class-specific precision/recall provide a more comprehensive assessment beyond accuracy [340]. These approaches could help address the challenges of prescriptive monitoring in settings with multi-class outcomes.

BIBLIOGRAPHY

References

[1] Albert Einstein. “Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]”. In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: `http://dx.doi.org/10.1002/andp.19053221004`.

[2] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

[3] Donald Knuth. *Knuth: Computers and Typesetting*. URL: `http://www-cs-faculty.stanford.edu/%5C~%7B%7Duno/abcde.html`.

[4] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A. Reijers. *Fundamentals of Business Process Management, Second Edition*. Springer, 2018.

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

[6] Kevin P. Murphy. *Machine learning - a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press, 2012.

[7] Michael I. Jordan. “Machine learning: trends, perspectives and challenges”. In: *TURC*. ACM, 2018, p. 8.

[8] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer Series in Statistics. Springer, 2009.

[9] Anna Leontjeva, Raffaele Conforti, Chiara Di Francescomarino, Marlon Dumas, and Fabrizio Maria Maggi. “Complex Symbolic Sequence Encodings for Predictive Monitoring of Business Processes”. In: *BPM*. Vol. 9253. Lecture Notes in Computer Science. Springer, 2015, pp. 297–313.

[10] Irene Teinemaa, Marlon Dumas, Marcello La Rosa, and Fabrizio Maria Maggi. “Outcome-Oriented Predictive Process Monitoring: Review and Benchmark”. In: *ACM Trans. Knowl. Discov. Data* 13.2 (2019), 17:1–17:57.

[11] Ping Wang, Yan Li, and Chandan K. Reddy. “Machine Learning for Survival Analysis: A Survey”. In: *ACM Comput. Surv.* 51.6 (2019), 110:1–110:36.

[12] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. “A Survey on Causal Inference”. In: *ACM Trans. Knowl. Discov. Data* 15.5 (2021), 74:1–74:46.

[13] Julian Schrittwieser, Thomas Hubert, Amol Mandhane, Mohammadamin Barekatain, Ioannis Antonoglou, and David Silver. “Online and Of-

fline Reinforcement Learning by Planning with a Learned Model". In: *NeurIPS*. 2021, pp. 27580–27591.

[14] Mirko Polato, Alessandro Sperduti, Andrea Burattin, and Massimiliano de Leoni. "Time and activity sequence prediction of business process instances". In: *Computing* 100.9 (2018), pp. 1005–1031.

[15] Massimiliano de Leoni, Wil M. P. van der Aalst, and Marcus Dees. "A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs". In: *Inf. Syst.* 56 (2016), pp. 235–257.

[16] Sjoerd van der Spoel, Maurice van Keulen, and Chintan Amrit. "Process Prediction in Noisy Data Sets: A Case Study in a Dutch Hospital". In: *SIMPDA*. Vol. 162. Lecture Notes in Business Information Processing. Springer, 2012, pp. 60–83.

[17] Anna Leontjeva, Raffaele Conforti, Chiara Di Francescomarino, Marlon Dumas, and Fabrizio Maria Maggi. "Complex Symbolic Sequence Encodings for Predictive Monitoring of Business Processes". In: *BPM*. Vol. 9253. Lecture Notes in Computer Science. Springer, 2015, pp. 297–313.

[18] Ilya Verenich, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi, and Irene Teinemaa. "Survey and Cross-benchmark Comparison of Remaining Time Prediction Methods in Business Process Monitoring". In: *ACM Trans. Intell. Syst. Technol.* 10.4 (2019), 34:1–34:34.

[19] Leo Breiman. "Random Forests". In: *Mach. Learn.* 45.1 (2001), pp. 5–32.

[20] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *KDD*. ACM, 2016, pp. 785–794.

[21] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics* (2001), pp. 1189–1232.

[22] Liudmila Ostroumova Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. "CatBoost: unbiased boosting with categorical features". In: *NeurIPS*. 2018, pp. 6639–6649.

[23] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *NIPS*. 2017, pp. 3146–3154.

[24] Andrew P. Bradley. "The use of the area under the ROC curve in the evaluation of machine learning algorithms". In: *Pattern Recognit.* 30.7 (1997), pp. 1145–1159.

[25] Miguel A Hernán and James M Robins. *Causal Inference: What If*. Boca Raton: Chapman & Hall/CRC, 2020.

[26] Judea Pearl. "Causal Inference". In: *NIPS Causality: Objectives and Assessment*. Vol. 6. JMLR Proceedings. JMLR.org, 2010, pp. 39–58.

[27] Donald B Rubin. “Estimating causal effects of treatments in randomized and nonrandomized studies.” In: *Journal of educational Psychology* 66.5 (1974), p. 688.

[28] Paul W Holland. “Statistics and causal inference”. In: *Journal of the American statistical Association* 81.396 (1986), pp. 945–960.

[29] Susan Athey, Guido W Imbens, Jonas Metzger, and Evan Munro. “Using wasserstein generative adversarial networks for the design of monte carlo simulations”. In: *Journal of Econometrics* (2021), p. 105076.

[30] Brady Neal, Chin-Wei Huang, and Sunand Raghupathi. “Realcause: Realistic causal inference benchmarking”. In: *arXiv preprint arXiv:2011.15007* (2020).

[31] Uri Shalit, Fredrik D. Johansson, and David A. Sontag. “Estimating individual treatment effect: generalization bounds and algorithms”. In: *ICML*. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 3076–3085.

[32] Christiana Kartsonaki. “Survival analysis”. In: *Diagnostic Histopathology* 22.7 (2016), pp. 263–270.

[33] Manish Kumar Goel, Pardeep Khanna, and Jugal Kishore. “Understanding survival analysis: Kaplan-Meier estimate”. In: *International journal of Ayurveda research* 1.4 (2010), p. 274.

[34] John Fox and Sanford Weisberg. “Cox proportional-hazards regression for survival data”. In: *An R and S-PLUS companion to applied regression* 2002 (2002).

[35] David G Kleinbaum, Mitchel Klein, David G Kleinbaum, and Mitchel Klein. “Parametric survival models”. In: *Survival Analysis: A Self-Learning Text* (2012), pp. 289–361.

[36] Brandon George, Samantha Seals, and Inmaculada Aban. “Survival analysis and regression models”. In: *Journal of nuclear cardiology* 21.4 (2014), pp. 686–694.

[37] Andrey Malinin, Liudmila Prokhorenkova, and Aleksei Ustimenko. “Uncertainty in Gradient Boosting via Ensembles”. In: *ICLR*. OpenReview.net, 2021.

[38] Eyke Hüllermeier and Willem Waegeman. “Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods”. In: *Mach. Learn.* 110.3 (2021), pp. 457–506.

[39] Stephen C Hora. “Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management”. In: *Reliability Engineering & System Safety* 54.2-3 (1996), pp. 217–223.

[40] Armen Der Kiureghian and Ove Ditlevsen. “Aleatory or epistemic? Does it matter?” In: *Structural safety* 31.2 (2009), pp. 105–112.

[41] Glenn Shafer and Vladimir Vovk. “A Tutorial on Conformal Prediction”. In: *J. Mach. Learn. Res.* 9 (2008), pp. 371–421.

[42] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. “Reinforcement learning: A survey”. In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285.

[43] Hai Nguyen and Hung M. La. “Review of Deep Reinforcement Learning for Robot Manipulation”. In: *IRC*. IEEE, 2019, pp. 590–595.

[44] Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. “Efficient online reinforcement learning with offline data”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 1577–1594.

[45] Kateryna Kubrak, Fredrik Milani, Alexander Nolte, and Marlon Dumas. “Prescriptive process monitoring: *Quo vadis*?” In: *PeerJ Comput. Sci.* 8 (2022), e1097.

[46] Stephan A. Fahrenkrog-Petersen, Niek Tax, Irene Teinemaa, Marlon Dumas, Massimiliano de Leoni, Fabrizio Maria Maggi, and Matthias Weidlich. “Fire now, fire later: alarm-based systems for prescriptive process monitoring”. In: *Knowl. Inf. Syst.* 64.2 (2022), pp. 559–587.

[47] Irene Teinemaa, Niek Tax, Massimiliano de Leoni, Marlon Dumas, and Fabrizio Maria Maggi. “Alarm-Based Prescriptive Process Monitoring”. In: *BPM (Forum)*. Vol. 329. Lecture Notes in Business Information Processing. Springer, 2018, pp. 91–107.

[48] Andreas Metzger, Adrian Neubauer, Philipp Bohn, and Klaus Pohl. “Proactive Process Adaptation Using Deep Learning Ensembles”. In: *CAiSE*. Vol. 11483. Lecture Notes in Computer Science. Springer, 2019, pp. 547–562.

[49] Johny Ghattas, Pnina Soffer, and Mor Peleg. “Improving business process decision making based on past experience”. In: *Decis. Support Syst.* 59 (2014), pp. 93–107.

[50] Likewin Thomas, MV Manoj Kumar, and B Annappa. “Recommending an alternative path of execution using an online decision support system”. In: *Proceedings of the 2017 international conference on intelligent systems, metaheuristics & swarm intelligence*. 2017, pp. 108–112.

[51] Steven Mertens. “Personalized Process and Decision Support in Emergency Care by Mining Electronic Health Records and Sensor Data”. In: *PervasiveHealth*. ACM, 2020, pp. 415–418.

[52] Christian Haisjackl and Barbara Weber. “User Assistance during Process Execution - An Experimental Evaluation of Recommendation Strategies”. In: *Business Process Management Workshops*. Vol. 66. Lecture Notes in Business Information Processing. Springer, 2010, pp. 134–145.

[53] Raffaele Conforti, Massimiliano de Leoni, Marcello La Rosa, Wil M. P. van der Aalst, and Arthur H. M. ter Hofstede. “A recommendation system for predicting risks across multiple business process instances”. In: *Decis. Support Syst.* 69 (2015), pp. 1–19.

[54] Ivan Donadello, Chiara Di Francescomarino, Fabrizio Maria Maggi, Francesco Ricci, and Aladdin Shikhizada. “Outcome-Oriented Prescriptive Process Monitoring based on Temporal Logic Patterns”. In: *Eng. Appl. Artif. Intell.* 126 (2023), p. 106899.

[55] Christoph Gröger, Holger Schwarz, and Bernhard Mitschang. “Prescriptive Analytics for Recommendation-Based Business Process Optimization”. In: *BIS*. Vol. 176. Lecture Notes in Business Information Processing. Springer, 2014, pp. 25–37.

[56] Renuka Sindhgatta, Aditya K. Ghose, and Hoa Khanh Dam. “Context-Aware Analysis of Past Process Executions to Aid Resource Allocation Decisions”. In: *CAiSE*. Vol. 9694. Lecture Notes in Computer Science. Springer, 2016, pp. 575–589.

[57] Sven Weinzierl, Sebastian Dunzer, Sandra Zilker, and Martin Matzner. “Prescriptive Business Process Monitoring for Recommending Next Best Actions”. In: *BPM (Forum)*. Vol. 392. Lecture Notes in Business Information Processing. Springer, 2020, pp. 193–209.

[58] Sebastian Huber, Marian Fietta, and Sebastian Hof. “Next step recommendation and prediction based on process mining in adaptive case management”. In: *S-BPM ONE*. ACM, 2015, 3:1–3:9.

[59] Andreas Metzger, Tristan Kley, and Alexander Palm. “Triggering Proactive Business Process Adaptations via Online Reinforcement Learning”. In: *BPM*. Vol. 12168. Lecture Notes in Computer Science. Springer, 2020, pp. 273–290.

[60] Massimiliano de Leoni, Marcus Dees, and Laurens Reulink. “Design and Evaluation of a Process-aware Recommender System based on Prescriptive Analytics”. In: *ICPM*. IEEE, 2020, pp. 9–16.

[61] Zahra Dasht Bozorgi, Irene Teinemaa, Marlon Dumas, Marcello La Rosa, and Artem Polyvyanyy. “Prescriptive process monitoring based on causal effect estimation”. In: *Inf. Syst.* 116 (2023), p. 102198.

[62] Aekyung Kim, Josué Obregon, and Jae-Yoon Jung. “Constructing Decision Trees from Process Logs for Performer Recommendation”. In: *Business Process Management Workshops*. Vol. 171. Lecture Notes in Business Information Processing. Springer, 2013, pp. 224–236.

[63] Arif Wibisono, Amna Shifia Nisafani, Hyerim Bae, and You-Jin Park. “On-the-Fly Performance-Aware Human Resource Allocation in the Business Process Management Systems Environment Using Naive Bayes”. In: *AP-BPM*. Vol. 219. Lecture Notes in Business Information Processing. Springer, 2015, pp. 70–80.

[64] Josué Obregon, Aekyung Kim, and Jae-Yoon Jung. “DTMiner: A Tool for Decision Making Based on Historical Process Data”. In: *AP-BPM*. Vol. 159. Lecture Notes in Business Information Processing. Springer, 2013, pp. 81–91.

[65] Likewin Thomas, MV Manoj Kumar, et al. “An online decision support system for recommending an alternative path of execution”. In: *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE. 2017, pp. 1–9.

[66] Gyunam Park and Minseok Song. “Prediction-based Resource Allocation using LSTM and Minimum Cost and Maximum Flow Algorithm”. In: *ICPM*. IEEE, 2019, pp. 121–128.

[67] Joël Goossens, Tiblets Demewez, and Marwan Hassani. “Effective steering of customer journey via order-aware recommendation”. In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2018, pp. 828–837.

[68] Alessandro Terragni and Marwan Hassani. “Optimizing customer journey using process mining and sequence-aware recommendation”. In: *SAC*. ACM, 2019, pp. 57–65.

[69] Sven Weinzierl, Matthias Stierle, Sandra Zilker, and Martin Matzner. “A Next Click Recommender System for Web-based Service Analytics with Context-aware LSTMs”. In: *HICSS*. ScholarSpace, 2020, pp. 1–10.

[70] Zahra Dasht Bozorgi, Irene Teinemaa, Marlon Dumas, Marcello La Rosa, and Artem Polyvyanyy. “Prescriptive Process Monitoring for Cost-Aware Cycle Time Reduction”. In: *ICPM*. IEEE, 2021, pp. 96–103.

[71] Eberhard Heber, Holger Hagen, and Martin Schmollinger. “Application of process mining for improving adaptivity in case management systems”. In: *DEC*. Vol. P-244. LNI. GI, 2015, pp. 221–231.

[72] Kimon Batoulis. “Proactive Decision Support During Business Process Execution”. In: *EMoV+MinoPro@Modellierung*. Vol. 1185. CEUR Workshop Proceedings. CEUR-WS.org, 2014, pp. 35–41.

[73] Joyce Nakatumba, Michael Westergaard, and Wil MP van der Aalst. “A meta-model for operational support”. In: *BPM Center Report BPM-12-05, BPMcenter. org* (2012), pp. 16–32.

[74] Sen Yang, Xin Dong, Leilei Sun, Yichen Zhou, Richard A. Farneth, Hui Xiong, Randall S. Burd, and Ivan Marsic. “A Data-driven Process Recommender Framework”. In: *KDD*. ACM, 2017, pp. 2111–2120.

[75] Silvana Pereira Detro, Eduardo Alves Portela Santos, Hervé Panetto, Eduardo de Freitas Rocha Loures, Mario Lezoche, and Claudia Maria Cabral Moro Barra. “Applying process mining and semantic reasoning for process model customisation in healthcare”. In: *Enterp. Inf. Syst.* 14.7 (2020), pp. 983–1009.

[76] Irene Barba, Barbara Weber, and Carmelo Del Valle. “Supporting the Optimized Execution of Business Processes through Recommendations”. In: *Business Process Management Workshops (1)*. Vol. 99. Lecture Notes in Business Information Processing. Springer, 2011, pp. 135–140.

[77] Sameh Triki, Narjès Bellamine Ben Saoud, Julie Dugdale, and Chihab Hanachi. "Coupling Case Based Reasoning and Process Mining for a Web Based Crisis Management Decision Support System". In: *WETICE*. IEEE Computer Society, 2013, pp. 245–252.

[78] Nada Mohammed Abdulhameed, Iman MA Helal, Ahmed Awad, and Ehab Ezat. "A resource recommendation approach based on co-working history". In: *International Journal of Advanced Computer Science and Applications* 9.7 (2018).

[79] Michael Arias, Jorge Munoz-Gama, and Marcos Sepúlveda. "A Multi-criteria Approach for Team Recommendation". In: *Business Process Management Workshops*. Vol. 281. Lecture Notes in Business Information Processing. 2016, pp. 384–396.

[80] Michael Arias, Rodrigo Saavedra, Maira R Marques, Jorge Munoz-Gama, and Marcos Sepúlveda. "Human resource allocation in business process management and process mining: A systematic mapping study". In: *Management Decision* 56.2 (2018), pp. 376–405.

[81] Hamid R. Motahari Nezhad and Claudio Bartolini. "Next Best Step and Expert Recommendation for Collaborative Processes in IT Service Management". In: *BPM*. Vol. 6896. Lecture Notes in Computer Science. Springer, 2011, pp. 50–61.

[82] Jelmer Jan Koorn, Xixi Lu, Henrik Leopold, and Hajo A. Reijers. "Looking for Meaning: Discovering Action-Response-Effect Patterns in Business Processes". In: *BPM*. Vol. 12168. Lecture Notes in Computer Science. Springer, 2020, pp. 167–183.

[83] Jelmer Jan Koorn, Xixi Lu, Henrik Leopold, Niels Martin, Sam Verboven, and Hajo A. Reijers. "Mining Statistical Relations for Better Decision Making in Healthcare Processes". In: *ICPM*. IEEE, 2022, pp. 32–39.

[84] Jelmer Jan Koorn, Xixi Lu, Henrik Leopold, and Hajo A. Reijers. "From action to response to effect: Mining statistical relations in work processes". In: *Inf. Syst.* 109 (2022), p. 102035.

[85] Zahra Dasht Bozorgi, Irene Teinemaa, Marlon Dumas, Marcello La Rosa, and Artem Polyvyanyy. "Process Mining Meets Causal Machine Learning: Discovering Causal Rules from Event Logs". In: *ICPM*. IEEE, 2020, pp. 129–136.

[86] Hans Weytjens, Wouter Verbeke, and Jochen De Weerdt. "Timed Process Interventions: Causal Inference vs. Reinforcement Learning". In: *Business Process Management Workshops*. Vol. 492. Lecture Notes in Business Information Processing. Springer, 2023, pp. 245–258.

[87] Alessandro Padella, Massimiliano de Leoni, Onur Dogan, and Riccardo Galanti. "Explainable Process Prescriptive Analytics". In: *ICPM*. IEEE, 2022, pp. 16–23.

[88] Nijat Mehdiyev and Peter Fettke. “Prescriptive Process Analytics with Deep Learning and Explainable Artificial Intelligence”. In: *ECIS*. 2020.

[89] Anna L. Trella, Kelly W. Zhang, Inbal Nahum-Shani, Vivek Shetty, Finale Doshi-Velez, and Susan A. Murphy. “Designing Reinforcement Learning Algorithms for Digital Interventions: Pre-Implementation Guidelines”. In: *Algorithms* 15.8 (2022), p. 255.

[90] Andreas Metzger, Clément Quinton, Zoltán Ádám Mann, Luciano Baresi, and Klaus Pohl. “Feature Model-Guided Online Reinforcement Learning for Self-Adaptive Services”. In: *ICSOC*. Vol. 12571. Lecture Notes in Computer Science. Springer, 2020, pp. 269–286.

[91] Andreas Metzger, Johannes Franke, and Thomas Jansen. “Ensemble Deep Learning for Proactive Terminal Process Management at the Port of Duisburg "duisport"”. In: *Business Process Management Cases (2)*. Springer, 2021, pp. 153–164.

[92] Andreas Metzger, Tristan Kley, Aristide Rothweiler, and Klaus Pohl. “Automatically reconciling the trade-off between prediction accuracy and earliness in prescriptive business process monitoring”. In: *Inf. Syst.* 118 (2023), p. 102254.

[93] Alexandros Bousdekis, Athanasios Kerasiotis, Silvester Kotsias, Georgia Theodoropoulou, Georgios Miaoulis, and Djamchid Ghazanfarpour. “Modelling and predictive monitoring of business processes under uncertainty with reinforcement learning”. In: *Sensors* 23.15 (2023), p. 6931.

[94] Silvester Kotsias, Athanasios Kerasiotis, Alexandros Bousdekis, Georgia Theodoropoulou, and Georgios Miaoulis. “Predictive and Prescriptive Business Process Monitoring with Reinforcement Learning”. In: *NiDS*. Vol. 556. Lecture Notes in Networks and Systems. Springer, 2022, pp. 245–254.

[95] Andrea Chiorrini, Claudia Diamantini, Alex Mircoli, and Domenico Potena. “A Preliminary Study on the Application of Reinforcement Learning for Predictive Process Monitoring”. In: *ICPM Workshops*. Vol. 406. Lecture Notes in Business Information Processing. Springer, 2020, pp. 124–135.

[96] Marco Pegoraro, Merih Seran Uysal, and Wil MP van der Aalst. “Discovering process models from uncertain event data”. In: *International Conference on Business Process Management*. Springer. 2019, pp. 238–249.

[97] Marco Pegoraro and Wil MP van der Aalst. “Mining uncertain event data in process mining”. In: *2019 International Conference on Process Mining (ICPM)*. IEEE. 2019, pp. 89–96.

[98] Marco Pegoraro, Merih Seran Uysal, and Wil MP van der Aalst. “Efficient time and space representation of uncertain event data”. In: *Algorithms* 13.11 (2020), p. 285.

[99] Chiara Di Francescomarino, Chiara Ghidini, Fabrizio Maria Maggi, and Fredrik Milani. “Predictive Process Monitoring Methods: Which One Suits Me Best?” In: *BPM*. Vol. 11080. Lecture Notes in Computer Science. Springer, 2018, pp. 462–479.

[100] Wil MP van der Aalst and Josep Carmona. *Process mining handbook*. Springer Nature, 2022.

[101] Vincenzo Pasquadibisceglie, Annalisa Appice, Giovanna Castellano, and Donato Malerba. “JARVIS: Joining Adversarial Training With Vision Transformers in Next-Activity Prediction”. In: *IEEE Transactions on Services Computing* (2023).

[102] Jari Peeperkorn, Seppe vanden Broucke, and Jochen De Weerdt. “Validation set sampling strategies for predictive process monitoring”. In: *Inf. Syst.* 121 (2024), p. 102330.

[103] Muhammad Awais Ali, Marlon Dumas, and Fredrik Milani. “Enhancing the Accuracy of Predictors of Activity Sequences of Business Processes”. In: *RCIS (1)*. Vol. 513. Lecture Notes in Business Information Processing. Springer, 2024, pp. 149–165.

[104] Na Guo, Cong Liu, Caihong Li, Qingtian Zeng, Chun Ouyang, Qingzhi Liu, and Xixi Lu. “Explainable and Effective Process Remaining Time Prediction Using Feature-Informed Cascade Prediction Model”. In: *IEEE Trans. Serv. Comput.* 17.3 (2024), pp. 949–962.

[105] Johannes De Smedt and Jochen De Weerdt. “Predictive process model monitoring using long short-term memory networks”. In: *Eng. Appl. Artif. Intell.* 133 (2024), p. 108295.

[106] Annina Liessmann, Weixin Wang, Sven Weinzierl, Sandra Zilker, and Martin Matzner. “Transfer Learning for Predictive Process Monitoring”. In: (2024).

[107] Jan Niklas Adams, Hannes Drescher, Andreas Swoboda, Nikou Günnemann, Gyunam Park, and Wil van der Aalst. “Improving Predictive Process Monitoring Using Object-Centric Process Mining”. In: (2024).

[108] Thaddeus Kosciuszek and Marwan Hassani. “Online Next Activity Prediction Under Concept Drifts”. In: *CAiSE Workshops*. Vol. 521. Lecture Notes in Business Information Processing. Springer, 2024, pp. 335–346.

[109] Anjo Seidel, Stephan Haarmann, and Mathias Weske. “Model-Based Recommendations for Next-Best Actions in Knowledge-Intensive Processes”. In: *CAiSE*. Vol. 14663. Lecture Notes in Computer Science. Springer, 2024, pp. 195–211.

[110] Stephen Pauwels and Toon Calders. “Incremental Predictive Process Monitoring: The Next Activity Case”. In: *BPM*. Vol. 12875. Lecture Notes in Computer Science. Springer, 2021, pp. 123–140.

[111] Christopher Klinkmüller, N. R. T. P. van Beest, and Ingo Weber. “Towards Reliable Predictive Process Monitoring”. In: *CAiSE Forum*. Vol. 317. Lec-

ture Notes in Business Information Processing. Springer, 2018, pp. 163–181.

[112] Williams Rizzi, Chiara Di Francescomarino, and Fabrizio Maria Maggi. "Explainability in Predictive Process Monitoring: When Understanding Helps Improving". In: *BPM (Forum)*. Vol. 392. Lecture Notes in Business Information Processing. Springer, 2020, pp. 141–158.

[113] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul W. Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Inf. Fusion* 76 (2021), pp. 243–297.

[114] Apostolos F. Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. "Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons". In: *J. Comput. Phys.* 477 (2023), p. 111902.

[115] Brian Jalaian and Stephen Russell. "Uncertain Context: Uncertainty Quantification in Machine Learning". In: *AI Mag.* 40.4 (2019), pp. 40–49.

[116] Hans Weytjens and Jochen De Weerdt. "Learning Uncertainty with Artificial Neural Networks for Improved Remaining Time Prediction of Business Processes". In: *BPM*. Vol. 12875. Lecture Notes in Computer Science. Springer, 2021, pp. 141–157.

[117] Pietro Portolani, Alessandro Brusaferri, Andrea Ballarino, and Matteo Matteucci. "Uncertainty in Predictive Process Monitoring". In: *IPMU (2)*. Vol. 1602. Communications in Computer and Information Science. Springer, 2022, pp. 547–559.

[118] Ioannis Prasidis, Nikolaos-Paraskevas Theodoropoulos, Alexandros Bousdekis, Georgia Theodoropoulou, and Georgios Miaoulis. "Handling Uncertainty in Predictive Business Process Monitoring with Bayesian Networks". In: *IISA*. IEEE, 2021, pp. 1–8.

[119] Meelis Kull and Peter A. Flach. "Reliability Maps: A Tool to Enhance Probability Estimates and Improve Classification Accuracy". In: *ECML/PKDD (2)*. Vol. 8725. Lecture Notes in Computer Science. Springer, 2014, pp. 18–33.

[120] Anastasios N. Angelopoulos and Stephen Bates. "Conformal Prediction: A Gentle Introduction". In: *Found. Trends Mach. Learn.* 16.4 (2023), pp. 494–591.

[121] Haocheng Lei and Anthony Bellotti. "Reliable prediction intervals with directly optimized inductive conformal regression for deep learning". In: *Neural Networks* 168 (2023), pp. 194–205.

[122] Alex Kendall and Yarin Gal. “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?” In: *NIPS*. 2017, pp. 5574–5584.

[123] Kevin P. Murphy. *Machine learning - a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press, 2012.

[124] Thomas G. Dietterich. “Ensemble Methods in Machine Learning”. In: *Multiple Classifier Systems*. Vol. 1857. Lecture Notes in Computer Science. Springer, 2000, pp. 1–15.

[125] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles”. In: *NIPS*. 2017, pp. 6402–6413.

[126] Tim Pearce, Alexandra Brintrup, Mohamed Zaki, and Andy Neely. “High-quality prediction intervals for deep learning: A distribution-free, ensembled approach”. In: *International conference on machine learning*. PMLR. 2018, pp. 4075–4084.

[127] Matteo Fontana, Gianluca Zeni, and Simone Vantini. “Conformal prediction: a unified review of theory and new challenges”. In: *Bernoulli* 29.1 (2023), pp. 1–23.

[128] Glenn Shafer and Vladimir Vovk. “A Tutorial on Conformal Prediction”. In: *J. Mach. Learn. Res.* 9 (2008), pp. 371–421.

[129] Martim Sousa. “Inductive Conformal Prediction: A Straightforward Introduction with Examples in Python”. In: *CoRR* abs/2206.11810 (2022).

[130] James Law. “Review of "Algorithmic Learning in a Random World by Vovk, Gammerman and Shafer", Springer, 2005, ISBN: 0-387-00152-2”. In: *SIGACT News* 37.4 (2006), pp. 38–40.

[131] Fotios Skouvas, Harris Papadopoulos, and Andreas S. Andreou. “Enhancing Predictive Process Monitoring with Conformal Prediction”. In: *AIAI (2)*. Vol. 712. IFIP Advances in Information and Communication Technology. Springer, 2024, pp. 201–214.

[132] Stefan Haas and Eyke Hüllermeier. “Conformalized prescriptive machine learning for uncertainty-aware automated decision making: the case of goodwill requests”. In: *International Journal of Data Science and Analytics* (2024), pp. 1–17.

[133] Lisa Jöckel, Michael Kläs, Janek Groß, and Pascal Gerber. “Conformal Prediction and Uncertainty Wrapper: What Statistical Guarantees Can You Get for Uncertainty Quantification in Machine Learning?” In: *SAFECOMP Workshops*. Vol. 14182. Lecture Notes in Computer Science. Springer, 2023, pp. 314–327.

[134] Bart F. A. Hompes, Abderrahmane Maaradji, Marcello La Rosa, Marlon Dumas, Joos C. A. M. Buijs, and Wil M. P. van der Aalst. “Discovering Causal Factors Explaining Business Process Performance Variation”. In:

CAiSE. Vol. 10253. Lecture Notes in Computer Science. Springer, 2017, pp. 177–192.

[135] Artem Polyvyanyy, Anastasiia Pika, Moe T Wynn, and Arthur HM Ter Hofstede. “A systematic approach for discovering causal dependencies between observations and incidents in the health and safety domain”. In: *Safety science* 118 (2019), pp. 345–354.

[136] Tanmayee Narendra, Prerna Agarwal, Monika Gupta, and Sampath Dechu. “Counterfactual Reasoning for Process Optimization Using Structural Causal Models”. In: *BPM Forum*. Vol. 360. Lecture Notes in Business Information Processing. Springer, 2019, pp. 91–106.

[137] Mahnaz Sadat Qafari and Wil M. P. van der Aalst. “Root Cause Analysis in Process Mining Using Structural Equation Models”. In: *Business Process Management Workshops*. Vol. 397. Lecture Notes in Business Information Processing. Springer, 2020, pp. 155–167.

[138] Mahnaz Sadat Qafari and Wil M. P. van der Aalst. “Case Level Counterfactual Reasoning in Process Mining”. In: *CAiSE Forum*. Vol. 424. Lecture Notes in Business Information Processing. Springer, 2021, pp. 55–63.

[139] Weijia Zhang, Jiuyong Li, and Lin Liu. “A Unified Survey of Treatment Effect Heterogeneity Modelling and Uplift Modelling”. In: *ACM Comput. Surv.* 54.8 (2022), 162:1–162:36.

[140] Vincent Dorie, George Perrett, Jennifer L. Hill, and Benjamin Goodrich. “Stan and BART for Causal Inference: Estimating Heterogeneous Treatment Effects Using the Power of Stan and the Flexibility of Machine Learning”. In: *Entropy* 24.12 (2022), p. 1782.

[141] Kathleen Kane, Victor SY Lo, and Jane Zheng. “Mining for the truly responsive customers and prospects using true-lift modeling: Comparison of new and existing methods”. In: *Journal of Marketing Analytics* 2 (2014), pp. 218–238.

[142] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. “Metalearners for estimating heterogeneous treatment effects using machine learning”. In: *Proceedings of the national academy of sciences* 116.10 (2019), pp. 4156–4165.

[143] Xinkun Nie and Stefan Wager. “Quasi-oracle estimation of heterogeneous treatment effects”. In: *Biometrika* 108.2 (2021), pp. 299–319.

[144] Susan Athey and Guido W Imbens. *Machine learning for estimating heterogeneous causal effects*. Tech. rep. 2015.

[145] Christos Louizos, Uri Shalit, Joris M. Mooij, David A. Sontag, Richard S. Zemel, and Max Welling. “Causal Effect Inference with Deep Latent-Variable Models”. In: *NIPS*. 2017, pp. 6446–6456.

[146] Piotr Rzepakowski and Szymon Jaroszewicz. “Decision trees for uplift modeling with single and multiple treatments”. In: *Knowl. Inf. Syst.* 32.2 (2012), pp. 303–327.

[147] Lukasz Zaniewicz and Szymon Jaroszewicz. “Support Vector Machines for Uplift Modeling”. In: *ICDM Workshops*. IEEE Computer Society, 2013, pp. 131–138.

[148] Stefan Wager and Susan Athey. “Estimation and inference of heterogeneous treatment effects using random forests”. In: *Journal of the American Statistical Association* 113.523 (2018), pp. 1228–1242.

[149] Miruna Oprescu, Vasilis Syrgkanis, and Zhiwei Steven Wu. “Orthogonal Random Forest for Causal Inference”. In: *ICML*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 4932–4941.

[150] Chathurangi Shyalika, Thushari P. Silva, and Asoka S. Karunananda. “Reinforcement Learning in Dynamic Task Scheduling: A Review”. In: *SN Comput. Sci.* 1.6 (2020), p. 306.

[151] Zhengxing Huang, Wil M. P. van der Aalst, Xudong Lu, and Huilong Duan. “Reinforcement learning based resource allocation in business process management”. In: *Data Knowl. Eng.* 70.1 (2011), pp. 127–145.

[152] Thais Rodrigues Neubauer, Valdinei Freire da Silva, Marcelo Fantinato, and Sarajane Marques Peres. “Resource Allocation Optimization in Business Processes Supported by Reinforcement Learning and Process Mining”. In: *BRACIS (1)*. Vol. 13653. Lecture Notes in Computer Science. Springer, 2022, pp. 580–595.

[153] Kamil Zbikowski, Michal Ostapowicz, and Piotr Gawrysiak. “Deep Reinforcement Learning for Resource Allocation in Business Processes”. In: *CoRR* abs/2104.00541 (2021).

[154] Ghada Soliman, Kareem Mostafa, and Omar Younis. “Reinforcement Learning for Process Mining: Business Process Optimization”. In: *World Conference on Information Systems and Technologies*. Springer. 2024, pp. 108–125.

[155] Iman Firouzian, Morteza Zahedi, and Hamid Hassanpour. “Cycle time optimization of processes using an entropy-based learning for task allocation”. In: *International Journal of Engineering* 32.8 (2019), pp. 1090–1100.

[156] Xingmei Liu, Jian Chen, Yu Ji, and Yang Yu. “Q-learning algorithm for task allocation based on social relation”. In: *Process-Aware Systems: First International Workshop, PAS 2014, Shanghai, China, October 17, 2014. Proceedings 1*. Springer. 2015, pp. 49–58.

[157] Alexandros Bousdekis, Athanasios Kerasiotis, Silvester Kotsias, Georgia Theodoropoulou, Georgios Miaoulis, and Djamchid Ghazanfarpour. “Modelling and Predictive Monitoring of Business Processes under

Uncertainty with Reinforcement Learning". In: *Sensors* 23.15 (2023), p. 6931.

[158] Hongming Zhang and Tianyang Yu. "Taxonomy of reinforcement learning algorithms". In: *Deep reinforcement learning: Fundamentals, research and applications* (2020), pp. 125–133.

[159] Peter Dayan and CJCH Watkins. "Q-learning". In: *Machine learning* 8.3 (1992), pp. 279–292.

[160] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous Methods for Deep Reinforcement Learning". In: *ICML*. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 1928–1937.

[161] Yuhui Wang, Hao He, and Xiaoyang Tan. "Truly proximal policy optimization". In: *Uncertainty in artificial intelligence*. PMLR. 2020, pp. 113–122.

[162] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. "Trust region policy optimization". In: *International conference on machine learning*. PMLR. 2015, pp. 1889–1897.

[163] Jeffrey D. Johnson, Jinghong Li, and Zengshi Chen. "Reinforcement Learning: An Introduction: R.S. Sutton, A.G. Barto, MIT Press, Cambridge, MA 1998, 322 pp. ISBN 0-262-19398-1". In: *Neurocomputing* 35.1-4 (2000), pp. 205–206.

[164] Stefano Branchi, Chiara Di Francescomarino, Chiara Ghidini, David Massimo, Francesco Ricci, and Massimiliano Ronzani. "Learning to Act: A Reinforcement Learning Approach to Recommend the Best Next Activities". In: *BPM (Forum)*. Vol. 458. Lecture Notes in Business Information Processing. Springer, 2022, pp. 137–154.

[165] Yuhui Wang, Hao He, and Xiaoyang Tan. "Truly Proximal Policy Optimization". In: *UAI*. Vol. 115. Proceedings of Machine Learning Research. AUAI Press, 2019, pp. 113–122.

[166] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[167] Anna Leontjeva, Raffaele Conforti, Chiara Di Francescomarino, Marlon Dumas, and Fabrizio Maria Maggi. "Complex Symbolic Sequence Encodings for Predictive Monitoring of Business Processes". In: *BPM*. Vol. 9253. Lecture Notes in Computer Science. Springer, 2015, pp. 297–313.

[168] Arik Senderovich, Matthias Weidlich, Avigdor Gal, and Avishai Mandelbaum. "Queue mining for delay prediction in multi-class service processes". In: *Inf. Syst.* 53 (2015), pp. 278–295.

[169] John L. Gustafson. "Little's Law". In: *Encyclopedia of Parallel Computing*. Springer, 2011, pp. 1038–1041.

[170] Chiara Di Francescomarino, Marlon Dumas, Fabrizio Maria Maggi, and Irene Teinemaa. "Clustering-Based Predictive Process Monitoring". In: *CoRR* abs/1506.01428 (2015).

[171] Tadayoshi Fushiki. "Estimation of prediction error by using K-fold cross-validation". In: *Statistics and Computing* 21 (2011), pp. 137–146.

[172] Li Yang and Abdallah Shami. "On hyperparameter optimization of machine learning algorithms: Theory and practice". In: *Neurocomputing* 415 (2020), pp. 295–316.

[173] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. "Algorithms for Hyper-Parameter Optimization". In: *NIPS*. 2011, pp. 2546–2554.

[174] Arik Senderovich, Chiara Di Francescomarino, Chiara Ghidini, Kerwin Jorbina, and Fabrizio Maria Maggi. "Intra and inter-case features in predictive process monitoring: A tale of two dimensions". In: *Business Process Management: 15th International Conference, BPM 2017, Barcelona, Spain, September 10–15, 2017, Proceedings 15*. Springer. 2017, pp. 306–323.

[175] Elizaveta Povalyaeva, Ismail Khamitov, and Artyom Fomenko. "Bpic 2017: density analysis of the interaction with clients". In: *BPI Challenge* 2017 (2017).

[176] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. "Scikit-learn: Machine Learning in Python". In: *J. Mach. Learn. Res.* 12 (2011), pp. 2825–2830.

[177] Scott M. Lundberg, Gabriel G. Erion, Hugh Chen, Alex J. DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. "From local explanations to global understanding with explainable AI for trees". In: *Nat. Mach. Intell.* 2.1 (2020), pp. 56–67.

[178] Randal S. Olson, William G. La Cava, Zairah Mustahsan, Akshay Varik, and Jason H. Moore. "Data-driven advice for applying machine learning to bioinformatics problems". In: *PSB*. 2018, pp. 192–203.

[179] Candice Bentéjac, Anna Csörgo, and Gonzalo Martínez-Muñoz. "A comparative analysis of gradient boosting algorithms". In: *Artif. Intell. Rev.* 54.3 (2021), pp. 1937–1967.

[180] Sjoerd van der Spoel, Maurice van Keulen, and Chintan Amrit. "Process Prediction in Noisy Data Sets: A Case Study in a Dutch Hospital". In:

SIMPDA. Vol. 162. Lecture Notes in Business Information Processing. Springer, 2012, pp. 60–83.

[181] Manuel Fernández Delgado, Eva Cernadas, Senén Barro, and Dinani Gomes Amorim. “Do we need hundreds of classifiers to solve real world classification problems?” In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 3133–3181.

[182] Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. “Classification with Valid and Adaptive Coverage”. In: *NeurIPS*. 2020.

[183] General Data Protection Regulation. “Art. 22 GDPR. Automated individual decision-making, including profiling”. In: *Intersoft Consulting, https://gdpr-info. eu/art-22-gdpr* (2020).

[184] Lihua Lei and Emmanuel J Candès. “Conformal inference of counterfactuals and individual treatment effects”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 83.5 (2021), pp. 911–938.

[185] Yong Yang, Marlon Dumas, Luciano García-Bañuelos, Artem Polyvyanyy, and Liang Zhang. “Generalized aggregate Quality of Service computation for composite services”. In: *J. Syst. Softw.* 85.8 (2012), pp. 1818–1830.

[186] Chiara Di Francescomarino and Chiara Ghidini. “Predictive Process Monitoring”. In: *Process Mining Handbook*. Vol. 448. Lecture Notes in Business Information Processing. Springer, 2022, pp. 320–346.

[187] Manuel Camargo, Marlon Dumas, and Oscar González Rojas. “Learning Accurate LSTM Models of Business Processes”. In: *BPM*. Vol. 11675. Lecture Notes in Computer Science. Springer, 2019, pp. 286–302.

[188] Li Lin, Lijie Wen, and Jianmin Wang. “MM-Pred: A Deep Predictive Model for Multi-attribute Event Sequence”. In: *SDM*. SIAM, 2019, pp. 118–126.

[189] Vincenzo Pasquadibisceglie, Annalisa Appice, Giovanna Castellano, and Donato Malerba. “Using Convolutional Neural Networks for Predictive Process Analytics”. In: *ICPM*. IEEE, 2019, pp. 129–136.

[190] Johannes De Smedt, Anton Yeshchenko, Artem Polyvyanyy, Jochen De Weerdt, and Jan Mendling. “Process model forecasting and change exploration using time series analysis of event sequence data”. In: *Data Knowl. Eng.* 145 (2023), p. 102145.

[191] Niek Tax, Ilya Verenich, Marcello La Rosa, and Marlon Dumas. “Predictive Business Process Monitoring with LSTM Neural Networks”. In: *CAiSE*. Vol. 10253. Lecture Notes in Computer Science. Springer, 2017, pp. 477–492.

[192] Farbod Taymouri, Marcello La Rosa, and Sarah M. Erfani. “A Deep Adversarial Model for Suffix and Remaining Time Prediction of Event Sequences”. In: *SDM*. SIAM, 2021, pp. 522–530.

[193] Marcus Dees, Massimiliano de Leoni, Wil M. P. van der Aalst, and Hajo A. Reijers. “What if process predictions are not followed by good recommendations?” In: *BPM (Industry Forum)*. Vol. 2428. CEUR Workshop Proceedings. CEUR-WS.org, 2019, pp. 61–72.

[194] Susan Athey. “Beyond prediction: Using big data for policy problems”. In: *Science* 355.6324 (2017), pp. 483–485.

[195] Kateryna Kubrak, Fredrik Milani, Alexander Nolte, and Marlon Dumas. “Design and Evaluation of a User Interface Concept for Prescriptive Process Monitoring”. In: *CAiSE*. Vol. 13901. Lecture Notes in Computer Science. Springer, 2023, pp. 347–363.

[196] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. “Design Science in Information Systems Research”. In: *MIS Q.* 28.1 (2004), pp. 75–105.

[197] Vagelis Plevris, German Solorzano, Nikolaos P Bakas, and Mohamed El Amine Ben Seghier. “Investigation of performance metrics in regression analysis and machine learning-based prediction models”. In: *8th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2022)*. European Community on Computational Methods in Applied Sciences. 2022.

[198] Michelangelo Ceci, Pasqua Fabiana Lanotte, Fabio Fumarola, Dario Pietro Cavallo, and Donato Malerba. “Completion Time and Next Activity Prediction of Processes Using Sequential Pattern Mining”. In: *Discovery Science*. Vol. 8777. Lecture Notes in Computer Science. Springer, 2014, pp. 49–61.

[199] Fernando Montoya, Hernán Astudillo, Daniela Díaz, and Esteban Berríos. “Causal Learning: Monitoring Business Processes Based on Causal Structures”. In: *Entropy* 26.10 (2024), p. 867.

[200] A. J. Figueredo and P. S. A. Wolf. “Assortative pairing and life history strategy - a cross-cultural study.” In: *Human Nature* 20 (2009), pp. 317–330.

[201] Brady Neal, Chin-Wei Huang, and Sunand Raghupathi. “RealCause: Realistic Causal Inference Benchmarking”. In: *CoRR* abs/2011.15007 (2020). arXiv: `2011.15007`. URL: `https://arxiv.org/abs/2011.15007`.

[202] Stephan A. Fahrenkrog-Petersen, Niek Tax, Irene Teinemaa, Marlon Dumas, Massimiliano de Leoni, Fabrizio Maria Maggi, and Matthias Weidlich. “Fire now, fire later: alarm-based systems for prescriptive process monitoring”. In: *Knowl. Inf. Syst.* 64.2 (2022), pp. 559–587.

[203] Andrey Malinin, Liudmila Prokhorenkova, and Aleksei Ustimenko. “Uncertainty in gradient boosting via ensembles”. In: *arXiv preprint arXiv:2006.10562* (2020).

[204] Chiara Di Francescomarino, Marlon Dumas, Fabrizio Maria Maggi, and Irene Teinemaa. “Clustering-based predictive process monitoring”. In: *IEEE Trans. Serv. Comput.* 12.6 (2016), pp. 896–909.

[205] Jongchan Kim, Marco Comuzzi, Marlon Dumas, Fabrizio Maria Maggi, and Irene Teinemaa. “Encoding resource experience for predictive process monitoring”. In: *Decis. Support Syst.* 153 (2022).

[206] Wolfgang Kratsch, Jonas Manderscheid, Maximilian Röglinger, and Johannes Seyfried. “Machine learning in business process monitoring: a comparison of deep learning and classical approaches used for outcome prediction”. In: *Bus. Inf. Syst. Eng.* 63.3 (2021), pp. 261–276.

[207] Yarin Gal and Zoubin Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *ICML*. PMLR. 2016, pp. 1050–1059.

[208] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. “A review of uncertainty quantification in deep learning: Techniques, applications and challenges”. In: *Information Fusion* 76 (2021), pp. 243–297.

[209] Prerna Agarwal, Avani Gupta, Renuka Sindhgatta, and Sampath Dechu. “Goal-Oriented Next Best Activity Recommendation using Reinforcement Learning”. In: *CoRR* abs/2205.03219 (2022).

[210] Sven Weinzierl, Sebastian Dunzer, Sandra Zilker, and Martin Matzner. “Prescriptive Business Process Monitoring for Recommending Next Best Actions”. In: *BPM (Forum)*. Vol. 392. Lecture Notes in Business Information Processing. Springer, 2020, pp. 193–209.

[211] Greg Van Houdt, Benoıt Depaire, and Niels Martin. “Root Cause Analysis in Process Mining with Probabilistic Temporal Logic”. In: ().

[212] Mahnaz Sadat Qafari and Wil MP van der Aalst. “Case level counterfactual reasoning in process mining”. In: *International Conference on Advanced Information Systems Engineering*. Springer. 2021, pp. 55–63.

[213] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. “Weight uncertainty in neural network”. In: *International conference on machine learning*. PMLR. 2015, pp. 1613–1622.

[214] Jeremy Nixon, Balaji Lakshminarayanan, and Dustin Tran. “Why Are Bootstrapped Deep Ensembles Not Better?” In: *”I Can’t Believe It’s Not Better!”NeurIPS 2020 workshop*. 2020.

[215] Liudmila Ostroumova Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. “CatBoost: unbiased boosting with categorical features”. In: *NeurIPS*. 2018.

[216] Aleksei Ustimenko and Liudmila Prokhorenkova. “SGLB: Stochastic Gradient Langevin Boosting”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 10487–10496.

[217] Eyke Hüllermeier and Willem Waegeman. “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods”. In: *Machine Learning* (2021).

[218] Stephen Pauwels and Toon Calders. “Incremental Predictive Process Monitoring: The Next Activity Case”. In: *BPM conference*. Springer. 2021, pp. 123–140.

[219] James V Stone. “Bayes’ rule: a tutorial introduction to Bayesian analysis”. In: (2013).

[220] Leif E Peterson. “K-nearest neighbor”. In: *Scholarpedia* 4.2 (2009), p. 1883.

[221] Fabrizio Maria Maggi, Chiara Di Francescomarino, Marlon Dumas, and Chiara Ghidini. “Predictive Monitoring of Business Processes”. In: *CAiSE*. Vol. 8484. Lecture Notes in Computer Science. Springer, 2014, pp. 457–472.

[222] Thomas G Dietterich et al. “Ensemble learning”. In: *The handbook of brain theory and neural networks* 2.1 (2002), pp. 110–125.

[223] Marlon Dumas, Manfred Reichert, and Ming-Chien Shan. *BPM 2008*. Vol. 5240. Springer, 2008.

[224] Mathias Weske. *Business process management architectures*. Springer, 2007.

[225] Arik Senderovich, Chiara Di Francescomarino, and Fabrizio Maria Maggi. “From knowledge-driven to data-driven inter-case feature encoding in predictive process monitoring”. In: *Inf. Syst.* 84 (2019), pp. 255–264.

[226] Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A Reijers, et al. *Fundamentals of business process management*. Vol. 1. Springer, 2018.

[227] Arik Senderovich, Chiara Di Francescomarino, and Fabrizio Maria Maggi. “From knowledge-driven to data-driven inter-case feature encoding in predictive process monitoring”. In: *Information Systems* 84 (2019), pp. 255–264.

[228] Irene Teinemaa, Marlon Dumas, Marcello La Rosa, and Fabrizio Maria Maggi. “Outcome-Oriented Predictive Process Monitoring: Review and Benchmark”. In: *ACM Trans. Knowl. Discov. Data* 13.2 (2019), 17:1–17:57.

[229] Joerg Evermann, Jana-Rebecca Rehse, and Peter Fettke. “Predicting process behaviour using deep learning”. In: *Decis. Support Syst.* 100 (2017), pp. 129–140.

[230] Ilya Verenich, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi, and Irene Teinemaa. “Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring”. In: *ACM TIST* 10.4 (2019), pp. 1–34.

[231] Anton Yeshchenko, Fernando Durier, Kate Revoredo, Jan Mendling, and Flavia Santoro. “Context-Aware Predictive Process Monitoring: The Impact of News Sentiment”. In: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer. 2018, pp. 586–603.

[232] Anna Leontjeva, Raffaele Conforti, Chiara Di Francescomarino, Marlon Dumas, and Fabrizio Maria Maggi. “Complex symbolic sequence encodings for predictive monitoring of business processes”. In: *International Conference on Business Process Management*. Springer. 2016, pp. 297–313.

[233] Sjoerd Van Der Spoel, Maurice Van Keulen, and Chintan Amrit. “Process prediction in noisy data sets: a case study in a dutch hospital”. In: *International Symposium on Data-Driven Process Discovery and Analysis*. Springer. 2012, pp. 60–83.

[234] Andreas Metzger and Felix Föcker. “Predictive Business Process Monitoring Considering Reliability Estimates”. In: *CAiSE*. Springer, 2017, pp. 445–460.

[235] Massimiliano De Leoni, Wil MP van der Aalst, and Marcus Dees. “A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs”. In: *Information Systems* 56 (2016), pp. 235–257.

[236] Wil MP Van der Aalst, Maja Pesic, and Minseok Song. “Beyond process mining: from the past to present and future”. In: *CAiSE*. 2010, pp. 38–52.

[237] Pnina Soffer, Johny Ghattas, and Mor Peleg. “A goal-based approach for learning in business processes”. In: *Intentional Perspectives on Information Systems Engineering*. Springer, 2010, pp. 239–256.

[238] Felix Mannhardt, Petter Arnesen, and Andreas D Landmark. “Estimating the impact of incidents on process delay”. In: *ICPM*. IEEE. 2019, pp. 49–56.

[239] Irene Teinemaa, Marlon Dumas, Anna Leontjeva, and Fabrizio Maria Maggi. “Temporal stability in predictive process monitoring”. In: *Data Min. Knowl. Discov.* 32.5 (2018), pp. 1306–1338.

[240] Tarun Tater, Sampath Dechu, Senthil Mani, and Chandresh Maurya. “Prediction of invoice payment status in account payable business process”. In: *International Conference on Service-Oriented Computing*. Springer. 2018, pp. 165–180.

[241] *4TU Centre for Research Data*. `https://researchdata.4tu.nl/`.

[242] Tom M Mitchell et al. *Machine learning*. 1997.

[243] Janez Demšar. “Statistical comparisons of classifiers over multiple data sets”. In: *Journal of Machine learning research* 7.Jan (2006), pp. 1–30.

[244] Arik Senderovich, Matthias Weidlich, Avigdor Gal, and Avishai Mandelbaum. "Queue mining for delay prediction in multi-class service processes". In: *Information Systems* 53 (2015), pp. 278–295.

[245] Eva L Klijn and Dirk Fahland. "Identifying and Reducing Errors in Remaining Time Prediction due to Inter-Case Dynamics". In: *2020 2nd International Conference on Process Mining (ICPM)*. IEEE. 2020, pp. 25–32.

[246] Niyi Ogunbiyi, Artie Basukoski, and Thierry Chaussalet. "Investigating Social Contextual Factors in Remaining-Time Predictive Process Monitoring—A Survival Analysis Approach". In: *Algorithms* 13.11 (2020), p. 267.

[247] John DC Little and Stephen C Graves. "Little's law". In: *Building intuition*. Springer, 2008, pp. 81–100.

[248] Guandong Xu, Tri Dung Duong, Qian Li, Shaowu Liu, and Xianzhi Wang. "Causality learning: A new perspective for interpretable machine learning". In: *arXiv preprint arXiv:2006.16789* (2020).

[249] Ruocheng Guo, Lu Cheng, Jundong Li, P Richard Hahn, and Huan Liu. "A survey of learning causality with data: Problems and methods". In: *ACM Comput. Surv.* 53.4 (2020), pp. 1–37.

[250] Stephan A Fahrenkrog-Petersen, Niek Tax, Irene Teinemaa, Marlon Dumas, Massimiliano de Leoni, Fabrizio Maria Maggi, and Matthias Weidlich. "Fire now, fire later: alarm-based systems for prescriptive process monitoring". In: *arXiv preprint arXiv:1905.09568* (2019).

[251] Andreas Metzger, Tristan Kley, and Alexander Palm. "Triggering Proactive Business Process Adaptations via Online Reinforcement Learning". In: *BPM*. Vol. 12168. Lecture Notes in Computer Science. Springer, 2020, pp. 273–290.

[252] Bart FA Hompes, Abderrahmane Maaradji, Marcello La Rosa, Marlon Dumas, Joos CAM Buijs, and Wil MP van der Aalst. "Discovering causal factors explaining business process performance variation". In: *International Conference on Advanced Information Systems Engineering*. Springer. 2017, pp. 177–192.

[253] Miruna Oprescu, Vasilis Syrgkanis, and Zhiwei Steven Wu. "Orthogonal random forest for causal inference". In: *ICML*. PMLR. 2019, pp. 4932–4941.

[254] Susan Athey, Julie Tibshirani, and Stefan Wager. "Generalized random forests". In: *The Annals of Statistics* 47.2 (2019), pp. 1148–1178.

[255] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. *Double/debiased machine learning for treatment and structural parameters*. 2018.

[256] Zahra Dasht Bozorgi, Irene Teinemaa, Marlon Dumas, Marcello La Rosa, and Artem Polyvyanyy. "Process Mining Meets Causal Machine Learn-

ing: Discovering Causal Rules from Event Logs". In: *2nd International Conference on Process Mining (ICPM)*. IEEE, 2020, pp. 129–136.

[257] Zahra Dasht Bozorgi, Irene Teinemaa, Marlon Dumas, Marcello La Rosa, and Artem Polyvyanyy. "Prescriptive Process Monitoring for Cost-Aware Cycle Time Reduction". In: *3rd International Conference on Process Mining (ICPM)*. IEEE, 2021, pp. 96–103.

[258] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. "Do we need hundreds of classifiers to solve real world classification problems?" In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 3133–3181.

[259] Hans Weytjens and Jochen De Weerdt. "Learning Uncertainty with Artificial Neural Networks for Improved Remaining Time Prediction of Business Processes". In: *International Conference on Business Process Management*. Springer. 2021, pp. 141–157.

[260] James Law. "Review of "Algorithmic Learning in a Random World by Vovk, Gammerman and Shafer", Springer, 2005, ISBN: 0-387-00152-2". In: *SIGACT News* 37.4 (2006), pp. 38–40.

[261] Kateryna Kubrak, Fredrik Milani, Alexander Nolte, and Marlon Dumas. "Prescriptive process monitoring: *Quo vadis*?" In: *PeerJ Comput. Sci.* 8 (2022), e1097.

[262] Christoph Gröger, Holger Schwarz, and Bernhard Mitschang. "Prescriptive Analytics for Recommendation-Based Business Process Optimization". In: *BIS*. Vol. 176. Lecture Notes in Business Information Processing. Springer, 2014, pp. 25–37.

[263] Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candès, and Aaditya Ramdas. "Conformal Prediction Under Covariate Shift". In: *NeurIPS*. 2019, pp. 2526–2536.

[264] Massimiliano de Leoni, Marcus Dees, and Laurens Reulink. "Design and Evaluation of a Process-aware Recommender System based on Prescriptive Analytics". In: *ICPM*. IEEE, 2020.

[265] Renuka Sindhgatta, Aditya K. Ghose, and Hoa Khanh Dam. "Context-Aware Analysis of Past Process Executions to Aid Resource Allocation Decisions". In: *CAiSE*. Vol. 9694. Lecture Notes in Computer Science. Springer, 2016, pp. 575–589.

[266] Volodya Vovk, Alexander Gammerman, and Craig Saunders. "Machine-Learning Applications of Algorithmic Randomness". In: *ICML*. Morgan Kaufmann, 1999, pp. 444–453.

[267] Gyunam Park and Minseok Song. "Prediction-based Resource Allocation using LSTM and Minimum Cost and Maximum Flow Algorithm". In: *ICPM*. IEEE, 2019, pp. 121–128.

[268] Glenn Shafer and Vladimir Vovk. "A Tutorial on Conformal Prediction". In: *J. Mach. Learn. Res.* 9 (2008), pp. 371–421.

[269] Wil Van Der Aalst. *Process mining: data science in action*. Vol. 2. Springer, 2016.

[270] Irene Teinemaa, Niek Tax, Massimiliano de Leoni, Marlon Dumas, and Fabrizio Maria Maggi. “Alarm-Based Prescriptive Process Monitoring”. In: *BPM (Forum)*. Vol. 329. Lecture Notes in Business Information Processing. Springer, 2018, pp. 91–107.

[271] Alfonso Eduardo Márquez-Chamorro, Manuel Resinas, and Antonio Ruiz-Cortés. “Predictive monitoring of business processes: a survey”. In: *IEEE Transactions on Services Computing* 11.6 (2017), pp. 962–977.

[272] Sven Weinzierl, Sebastian Dunzer, Sandra Zilker, and Martin Matzner. “Prescriptive Business Process Monitoring for Recommending Next Best Actions”. In: *BPM Forum*. Vol. 392. Springer, 2020, pp. 193–209.

[273] Hans Weytjens, Wouter Verbeke, and Jochen De Weerdt. “Timing Process Interventions with Causal Inference and Reinforcement Learning”. In: *CoRR* abs/2306.04299 (2023).

[274] Zahra Dasht Bozorgi, Irene Teinemaa, Marlon Dumas, Marcello La Rosa, and Artem Polyvyanyy. “Prescriptive process monitoring based on causal effect estimation”. In: *Information Systems* 116 (2023), p. 102198.

[275] Alexander Palm, Andreas Metzger, and Klaus Pohl. “Online Reinforcement Learning for Self-adaptive Information Systems”. In: *CAiSE*. Vol. 12127. Lecture Notes in Computer Science. Springer, 2020, pp. 169–184.

[276] Fadi Baskharon, Ahmed Awad, and Chiara Di Francescomarino. “Predicting Remaining Cycle Time from Ongoing Cases: A Survival Analysis-Based Approach”. In: *ICPM Workshops*. Vol. 406. Lecture Notes in Business Information Processing. Springer, 2020, pp. 99–111.

[277] Stefano Branchi, Andrei Buliga, Chiara Di Francescomarino, Chiara Ghidini, Francesca Meneghello, and Massimiliano Ronzani. “Recommending the optimal policy by learning to act from temporal data”. In: *CoRR* abs/2303.09209 (2023).

[278] Gyunam Park and Minseok Song. “Prediction-based Resource Allocation using LSTM and Minimum Cost and Maximum Flow Algorithm”. In: *ICPM*. IEEE, 2019, pp. 121–128.

[279] Andreas Metzger, Tristan Kley, Aristide Rothweiler, and Klaus Pohl. “Automatically Reconciling the Trade-off between Prediction Accuracy and Earliness in Prescriptive Business Process Monitoring”. In: *CoRR* abs/2307.05939 (2023).

[280] Ivan Donadello, Chiara Di Francescomarino, Fabrizio Maria Maggi, Francesco Ricci, and Aladdin Shikhizada. “Outcome-Oriented Prescriptive Process Monitoring Based on Temporal Logic Patterns”. In: *CoRR* abs/2211.04880 (2022).

[281] John L. Gustafson. “Little’s Law”. In: *Encyclopedia of Parallel Computing*. Springer, 2011, pp. 1038–1041.

[282] Niek Tax, Ilya Verenich, Marcello La Rosa, and Marlon Dumas. “Predictive Business Process Monitoring with LSTM Neural Networks”. In: *CAiSE*. Vol. 10253. Lecture Notes in Computer Science. Springer, 2017, pp. 477–492.

[283] John P Klein, Melvin L Moeschberger, et al. *Survival analysis: techniques for censored and truncated data*. Vol. 1230. Springer, 2003.

[284] Judea Pearl. “Causal Inference”. In: *NIPS Causality: Objectives and Assessment*. Vol. 6. JMLR Proceedings. JMLR.org, 2010, pp. 39–58.

[285] Zahra Dasht Bozorgi, Irene Teinemaa, Marlon Dumas, Marcello La Rosa, and Artem Polyvyanyy. “Prescriptive process monitoring based on causal effect estimation”. In: *Inf. Syst.* 116 (2023), p. 102198.

[286] Gianluca Zeni, Matteo Fontana, and Simone Vantini. “Conformal Prediction: a Unified Review of Theory and New Challenges”. In: *CoRR* abs/2005.07972 (2020).

[287] Vladimir Vovk. “Cross-conformal predictors”. In: *Ann. Math. Artif. Intell.* 74.1-2 (2015), pp. 9–28.

[288] Mohamed-Amine Chadi and Hajar Mousannif. “Understanding Reinforcement Learning Algorithms: The Progress from Basic Q-learning to Proximal Policy Optimization”. In: *CoRR* abs/2304.00026 (2023).

[289] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. “Proximal Policy Optimization Algorithms”. In: *CoRR* abs/1707.06347 (2017).

[290] Bart J. Verhoef and Xixi Lu. “Using Reinforcement Learning to Optimize Responses in Care Processes: A Case Study on Aggression Incidents”. In: *CoRR* abs/2310.00981 (2023).

[291] Haocheng Lei and Anthony Bellotti. “Reliable prediction intervals with directly optimized inductive conformal regression for deep learning”. In: *Neural Networks* 168 (2023), pp. 194–205.

[292] Martim Sousa. “Inductive Conformal Prediction: A Straightforward Introduction with Examples in Python”. In: *CoRR* abs/2206.11810 (2022).

[293] Haocheng Lei and Anthony Bellotti. “Reliable prediction intervals with directly optimized inductive conformal regression for deep learning”. In: *Neural Networks* 168 (2023), pp. 194–205.

[294] Anastasios N. Angelopoulos and Stephen Bates. “Conformal Prediction: A Gentle Introduction”. In: *Found. Trends Mach. Learn.* 16.4 (2023), pp. 494–591.

[295] Brady Neal, Chin-Wei Huang, and Sunand Raghupathi. “RealCause: Realistic Causal Inference Benchmarking”. In: *CoRR* abs/2011.15007 (2020).

[296] Hans Weytjens, Wouter Verbeke, and Jochen De Weerdt. “Timed Process Interventions: Causal Inference vs. Reinforcement Learning”. In: *Business Process Management Workshops*. Ed. by Jochen De Weerdt and Luise Pufahl. Springer Nature Switzerland, 2024.

[297] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. “Algorithms for Hyper-Parameter Optimization”. In: *NIPS*. 2011, pp. 2546–2554.

[298] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani B. Srivastava. “How Can I Explain This to You? An Empirical Study of Deep Neural Network Explanation Methods”. In: *NeurIPS*. 2020.

[299] René Michel, Igor Schnakenburg, and Tobias von Martens. “Targeting Uplift”. In: ().

[300] Alsane Sene, Bernard Kamsu-Foguem, and Pierre Rumeau. “Data mining for decision support with uncertainty on the airplane”. In: *Data Knowl. Eng.* 117 (2018), pp. 18–36.

[301] Leo Guelman, Montserrat Guillén, and Ana Maria Pérez-Marín. “A decision support framework to implement optimal personalized marketing interventions”. In: *Decis. Support Syst.* 72 (2015), pp. 24–32.

[302] Soroosh Nalchigar and Eric S. K. Yu. “Business-driven data analytics: A conceptual modeling framework”. In: *Data Knowl. Eng.* 117 (2018), pp. 359–372.

[303] Kees M. van Hee, Alexander Serebrenik, Natalia Sidorova, Marc Voorhoeve, and Jan van der Wal. “Scheduling-free resource management”. In: *Data Knowl. Eng.* 61.1 (2007), pp. 59–75.

[304] Marco Comuzzi, Jochem Vonk, and Paul W. P. J. Grefen. “Measures and mechanisms for process monitoring in evolving business networks”. In: *Data Knowl. Eng.* 71.1 (2012), pp. 1–28.

[305] Yuhee Kwon and Zoonky Lee. “A hybrid decision support system for adaptive trading strategies: Combining a rule-based expert system with a deep reinforcement learning strategy”. In: *Decis. Support Syst.* 177 (2024), p. 114100.

[306] James Todd, Adrian Gepp, Steven E. Stern, and Bruce James Vanstone. “Improving decision making in the management of hospital readmissions using modern survival analysis techniques”. In: *Decis. Support Syst.* 156 (2022), p. 113747.

[307] Judea Pearl. “Causal inference”. In: *Causality: objectives and assessment* (2010), pp. 39–58.

[308] Lukas Baltramaitis. “Activity-Oriented Causal Process Mining: An End-to-End Approach Utilizing Ylearn”. MA thesis. Tartu, Estonia: University of Tartu, 2022. URL: `https://dspace.ut.ee/items/5163b187-d35c-4092-9448-2a146eaed6cf`.

[309] Peter C Austin and Elizabeth A Stuart. "Moving towards best practice when using inverse probability of treatment weighting (IPTW) using the propensity score to estimate causal treatment effects in observational studies". In: *Statistics in medicine* 34.28 (2015), pp. 3661–3679.

[310] Christos E Papadopoulos and Hoi Yeung. "Uncertainty estimation and Monte Carlo simulation method". In: *Flow Measurement and Instrumentation* 12.4 (2001), pp. 291–298.

[311] Yechao Huang, Ting Shu, and Zuohua Ding. "A learn-to-rank method for model-based regression test case prioritization". In: *IEEE Access* 9 (2021), pp. 16365–16382.

[312] Michael L McManus, Michael C Long, Abbot Cooper, and Eugene Litvak. "Queuing theory accurately models the need for critical care resources". In: *The Journal of the American Society of Anesthesiologists* 100.5 (2004), pp. 1271–1276.

[313] Tim Pearce, Alexandra Brintrup, Mohamed Zaki, and Andy Neely. "High-Quality Prediction Intervals for Deep Learning: A Distribution-Free, Ensembled Approach". In: *ICML*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 4072–4081.

[314] Paolo Ceravolo, Sylvio Barbon Junior, Ernesto Damiani, and Wil M. P. van der Aalst. "Tuning Machine Learning to Address Process Mining Requirements". In: *IEEE Access* 12 (2024), pp. 24583–24595.

[315] Mahmoud Shoush and Marlon Dumas. "When to Intervene? Prescriptive Process Monitoring Under Uncertainty and Resource Constraints". In: *BPM (Forum)*. Vol. 458. Lecture Notes in Business Information Processing. Springer, 2022, pp. 207–223.

[316] Mahmoud Shoush and Marlon Dumas. "Intervening With Confidence: Conformal Prescriptive Monitoring of Business Processes". In: *PMAI@IJCAI*. Vol. 3569. CEUR Workshop Proceedings. CEUR-WS.org, 2023, pp. 1–12.

[317] Mahmoud Shoush and Marlon Dumas. "Prescriptive Process Monitoring Under Resource Constraints: A Causal Inference Approach". In: *ICPM Workshops*. Vol. 433. Lecture Notes in Business Information Processing. Springer, 2021, pp. 180–193.

[318] Denise Maria Vecino Sato, Sheila Cristiana De Freitas, Jean Paul Barddal, and Edson Emílio Scalabrin. "A Survey on Concept Drift in Process Mining". In: *ACM Comput. Surv.* 54.9 (2022), 189:1–189:38.

[319] Lucas Baier, Josua Reimold, and Niklas Kühl. "Handling Concept Drift for Predictions in Business Process Mining". In: *CBI (1)*. IEEE, 2020, pp. 76–83.

[320] Tsung-Hao Huang, Andreas Metzger, and Klaus Pohl. "Counterfactual Explanations for Predictive Business Process Monitoring". In: *CoRR* abs/2202.12018 (2022).

[321] Andrei Buliga, Chiara Di Francescomarino, Chiara Ghidini, Ivan Donadello, and Fabrizio Maria Maggi. "Guiding the generation of counterfactual explanations through temporal background knowledge for Predictive Process Monitoring". In: *CoRR* abs/2403.11642 (2024).

[322] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. "Transformers: State-of-the-Art Natural Language Processing". In: *EMNLP (Demos)*. Association for Computational Linguistics, 2020, pp. 38–45.

[323] Ajay Tanikonda, Brij Kishore Pandey, Sudhakar Reddy Peddinti, and Subba Rao Katragadda. "Application of Transformer Models for Advanced Process Optimization and Process Mining". In: *Journal of Science & Technology (JST)* 3.5 (2022).

[324] Alex Graves and Alex Graves. "Long short-term memory". In: *Supervised sequence labelling with recurrent neural networks* (2012), pp. 37–45.

[325] Niek Tax, Ilya Verenich, Marcello La Rosa, and Marlon Dumas. "Predictive business process monitoring with LSTM neural networks". In: *Advanced Information Systems Engineering: 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings 29*. Springer. 2017, pp. 477–492.

[326] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. "The graph neural network model". In: *IEEE transactions on neural networks* 20.1 (2008), pp. 61–80.

[327] Dominique Sommers, Vlado Menkovski, and Dirk Fahland. "Process discovery using graph neural networks". In: *2021 3rd International Conference on Process Mining (ICPM)*. IEEE. 2021, pp. 40–47.

[328] Manuel Camargo, Daniel Báron, Marlon Dumas, and Oscar González-Rojas. "Learning business process simulation models: a hybrid process mining and deep learning approach". In: *Information Systems* 117 (2023), p. 102248.

[329] Vincenzo Pasquadibisceglie, Annalisa Appice, Giovanna Castellano, and Donato Malerba. "Predictive process mining meets computer vision". In: *Business Process Management Forum: BPM Forum 2020, Seville, Spain, September 13–18, 2020, Proceedings 18*. Springer. 2020, pp. 176–192.

[330] Ekene Obodoekwe, Xianwen Fang, and Ke Lu. "Convolutional neural networks in process mining and data analytics for prediction accuracy". In: *Electronics* 11.14 (2022), p. 2128.

[331] Philippe Krajsic and Bogdan Franczyk. "Variational Autoencoder for Anomaly Detection in Event Data in Online Process Mining." In: *ICEIS (1)*. 2021, pp. 567–574.

[332] Dominic A Neu, Johannes Lahann, and Peter Fettke. “A systematic literature review on state-of-the-art deep learning methods for process prediction”. In: *Artificial Intelligence Review* 55.2 (2022), pp. 801–827.

[333] B Kamala and B Latha. “Process mining and deep neural network approach for the prediction of business process outcome”. In: *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*. IEEE. 2022, pp. 1–4.

[334] Francois Chollet. “The limitations of deep learning”. In: *Deep learning with Python* (2017).

[335] Vincenzo Pasquadibisceglie, Annalisa Appice, Giovanna Castellano, Donato Malerba, and Giuseppe Modugno. “ORANGE: outcome-oriented predictive process monitoring based on image encoding and CNNs”. In: *IEEE Access* 8 (2020), pp. 184073–184086.

[336] Eli N Weinstein and David M Blei. “Hierarchical causal models”. In: *arXiv preprint arXiv:2401.05330* (2024).

[337] Virginia Aglietti, Theodoros Damoulas, Mauricio Álvarez, and Javier González. “Multi-task causal learning with gaussian processes”. In: *Advances in neural information processing systems* 33 (2020), pp. 6293–6304.

[338] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[339] Jason Brownlee. “Cost-sensitive learning for imbalanced classification”. In: *Machine learning mastery* 17 (2020), p. 23.

[340] Margherita Grandini, Enrico Bagli, and Giorgio Visani. “Metrics for multi-class classification: an overview”. In: *arXiv preprint arXiv:2008.05756* (2020).

[341] Zahra Dasht Bozorgi, Marlon Dumas, Marcello La Rosa, Artem Polyvyanyy, **Mahmoud Shoush**, and Irene Teinemaa. “Learning When to Treat Business Processes: Prescriptive Process Monitoring with Causal Inference and Reinforcement Learning”. In: *Advanced Information Systems Engineering (CAiSE)*. Vol. 13901. Lecture Notes in Computer Science. Springer, 2023, 364–380.
DOI: `https://doi.org/10.1007/978-3-031-34560-9_22`.

[342] **Mahmoud Shoush** and Marlon Dumas. “White box specification of intervention policies for prescriptive process monitoring”. In: *Data & Knowledge Engineering* 155 (2024), 102379.
DOI: `https://doi.org/10.1016/j.datak.2024.102379`.

[343] **Mahmoud Shoush** and Marlon Dumas. “Prescriptive Process Monitoring Under Resource Constraints: A Reinforcement Learning Approach”. In: *KI-Künstliche Intelligenz* (2024), 1–22.
DOI: `https://doi.org/10.1007/s13218-024-00881-6`.

CODE REPOSITORIES

To ensure the reproducibility of the research results, we have made the implementations of the proposed approaches and the code for conducting the experiments reported in the thesis available in the following repository: https://github.com/mshoush/prescriptive-monitoring-PhD-thesis.

To maintain a consistent experimental framework across the thesis, the experiments presented here differ slightly from those in the original publications. The implementations and code used for the experiments in the original publications are available in the following repositories:

1. White box specification of intervention policies for prescriptive process monitoring: `https://github.com/mshoush/Prescriptive-Monitoring-Framework`.
2. Prescriptive Process Monitoring Under Resource Constraints: A Reinforcement Learning Approach: `https://github.com/mshoush/RL-prescriptive-monitoring`.
3. Intervening With Confidence: Conformal Prescriptive Monitoring of Business Processes: `https://github.com/mshoush/conformal-prescriptive-monitoring`.
4. When to Intervene? Prescriptive Process Monitoring Under Uncertainty and Resource Constraints: `https://github.com/mshoush/prescriptive-monitoring-uncertainty`.
5. Prescriptive Process Monitoring Under Resource Constraints: A Causal Inference Approach: `https://github.com/mshoush/Prescriptive-monitoring-Causal-Inference`.

ACKNOWLEDGEMENTS

All praise and thanks are due to Allah, the Creator of all that exists. Peace and blessings be upon His Prophet Muhammad (Peace be upon him), who guided humanity from darkness to light. I bear witness that there is no god worthy of worship except Allah, and I bear witness that His Prophet Muhammad is His servant and messenger.

I am extremely grateful to my supervisor, Prof. Marlon Dumas, for his invaluable guidance, patience, support, and feedback throughout my Ph.D. journey. His encouragement has been invaluable not only in completing this thesis but also in shaping my career and personal growth. I owe him tremendous gratitude for all I have learned from him.

I sincerely thank my thesis reviewers, Prof. Annalisa Appice, Assoc. Prof. Sylvio Barbon Junior, and Dr. Faiz Ali Shah, for their constructive feedback, which helped improve this work. My heartfelt appreciation also goes to the University of Tartu, Institute of Computer Science, for allowing me to pursue my PhD and providing a supportive academic environment. I am equally thankful to my friends and colleagues at the University of Tartu for their encouragement. Special thanks go to my colleague Anti Alman, who helped me with the Estonian translations in this thesis.

Special thanks to the Estonian Research Council (PRG1226), the European Research Council (PIX Project), and the city of Tartu for the resources and environment that supported this research. Additionally, we acknowledge the use of *ChatGPT*[1] to enhance the readability.

Finally, my heartfelt thanks go to my family for their unconditional love and support, which have been my greatest source of strength, particularly during my PhD years. I also express my deep gratitude to Prof. Salwa Nassar, Prof. Ahmed Awad, and the late Prof. Sherif Sakr for their guidance and belief in my potential, which has been pivotal to my academic journey.

Thank you All.

---

[1] OpenAI. (2025). *ChatGPT* [Large language model]. Available at: https://chat.openai.com

SISUKOKKUVÕTE

Ettekirjutav protsessijälgimine määramatuse ja ressursipiirangute korral

Kaasaegsetes organisatsioonides on korduvad tegevused koondatud äriprotsessideks, mida jälgitakse ja optimeeritakse pidevalt, et pakkuda klientidele ja teistele sidusrühmadele kõrgemat lisandväärtust. Protsessikaeve on tehnikate kogum, mis toetab äriprotsesside analüüsi ja optimeerimist tuginedes nende äriprotsesside täitmisel sündmuslogidesse kogutavatele andmetele. Alamosa protsessikaeve tehnikatest, nimetusega ennustav protsessijälgimine, kasutab neid sündmuslogisid, et treenida masinõppemudeleid, mis suudavad jooksvalt ennustada äriprotsessi tulevasi olekuid või sündmusi. Ennustava protsessijälgimise tehnikad võimaldavad juhtidel näiteks näha, millised protsessi täitmised omavad tõenäosust et protsessi tulemus on negatiivne, viies näiteks kliendi kaebusteni või müügivõimaluste kaotamiseni. Samas ei aita need tehnikad juhtidel otseselt kindlaks teha, milliste (täiendavate) tegevuste abil oleks võimalik neid negatiivseid tulemusi vältida.

Ettekirjutav protsessijälgimine on eelnevaga seotud tehnikate kogum, mis käsitleb ülaltoodud piirangut pakkudes välja sekkumisi, mis suurendavad protsessi positiivsete tulemuste tõenäosust. Näiteks, kui ennustav protsessijälgimine võib vaid tuvastada, et taotleja lükkab laenupakkumise tõenäoliselt tagasi, siis ettekirjutav protsessijälgimine võib lisaks soovitada, et laenuhaldur (ressurss) saadaks täiendava laenupakkumise (sekkumine) eesmärgiga suurendada laenulepingu sõlmimise tõenäosust (protsessi positiivne tulemus). Ettekirjutava protsessijälgimise selgrooks on sekkumispoliitika — otsustusprotseduur, mis määrab, millised protsessi isendid (juhud) nõuavad sekkumist, millise protsessioleku korral sekkumine algatada, ja milline ressurss peaks sekkumise teostama. Ettekirjutava protsessijälgimise eesmärk on maksimeerida võimendusfunktsiooni, mis arvestab nii negatiivsete tulemuste vältimisest saadavat kasu kui ka sekkumiste teostamisega kaasnevat lisakulu.

Ettekirjutava protsessi jälgimise valdkonna olemasolevatel lähenemistel on kaks tüüpilist piirangut. Esiteks ei arvesta nad soovituste aluseks olevate ennustuste määramatusega. Teiseks eeldavad nad iga soovitatud sekkumise teostamist, jättes arvestamata asjaolu, et sekkumise teostamine nõuab vastava ressursi aega, millest tulenevalt on võimalike sekkumiste koguarv piiratud. Teisisõnu, selles valdkonnas hetkel olemasolevad meetodid ei ole määramatuse- ega ressursiteadlikud.

See väitekiri esitleb kolme teaduslikku panust, mis keskenduvad nende piirangute lahendamisele. Esimene panus on ettekirjutav protsessijälgimise meetod, mis kaasab määramatuse näitajad sekkumispoliitikasse. Selle meetodi empiiriline hindamine näitab, et määramatuse näitajate kaasamine parandab sekkumispoliitika tulemuslikkust, eriti oludes kus ressursside kättesaadavus on tugevalt piiratud. Teiseks ja kolmandaks panuseks olevad ettekirjutava protsessijälgimise meetodid on määramatuse- ja ressursiteadlikud. Teine panus on reeglipõhine valge kasti lähenemine ettekirjutavale protsessijälgimisele. See meetod võimaldab kasutajatel määrata sekkumispoliitikaid filtreerimis- ja järjestuspoliitikate abil, tuginedes määramatust arvestavatele ennustus-, põhjuslikus-, ja ellujäämismudelite väljunditele. Kolmas panus on musta kasti lähenemine ettekirjutavale protsessijälgimisele. See meetod kasutab jooksvat stiimulõpet, et koostada sekkumispoliitika, mis ühendab ennustus-, põhjuslikus-, ja ellujäämismudelitest saadavad sisendid, et maksimeerida protsessi tulemuslikkuse võimendusfunktsiooni, ilma et kasutaja peaks ise sekkumisreegleid määrama.

Käesolevas väitekirjas eitatud meetodid on empiiriliselt hinnatud kasutades reaalseid andmestikke ning võttes võrdlusaluseks varasemad olemasolevad lähenemised, mis ei ole määramatuse- ega ressursiteadlikud. Hindamise tulemused näitavad, et sekkumispoliitikad on tõhusamad, kui nad kaasavad ennustus-, põhjuslikus-, ja ellujäämismudelite väljundeid ebakindlust teadvustaval viisil ja arvestavad ressursside kättesaadavusega ilmutatud kujul. Kõik antud väitekirja raames väljatöötatud meetodid on saadaval avatud lähtekoodiga tarkvarana, mis võimaldab teistel teadlastel seda tööd laiendada.

CURRICULUM VITAE

Personal data

Name:	Mahmoud Kamel Akila Soliman Shoush
Date of Birth:	07.04.1991
Citizenship:	Egyptian

Education

2020-2025	University of Tartu, Faculty of Science and Technology, doctoral studies, specialty: Computer Science.
2018-2020	University of Tartu, Faculty of Science and Technology, master studies, specialty: Data Science.
2010–2014	Menoufia University, Faculty of Electronic Engineering, bachelor studies, specialty: Computer Science and Engineering.

Employment

2024-	Cybernetica AS, Researcher
2020–2024	University of Tartu, Junior Research Fellow of Information Systems
2019–2020	University of Tartu, Data Scientist

Scientific work

Main fields of interest:

- machine learning
- process mining

ELULOOKIRJELDUS

Isikuandmed

Nimi: Mahmoud Kamel Akila Soliman Shoush
Sünniaeg: 07.04.1991
Kodakondsus: Egiptus

Haridus

2020-2025	Tartu Ülikool, Loodus- ja täppisteaduste valdkond, doktoriõpe, eriala: Informaatika.
2018-2020	Tartu Ülikool, Loodus- ja täppisteaduste valdkond, magistriõpe, eriala: Andmeteadus.
2010–2014	Menoufia Ülikool, elektroonika teaduskond, bakalaureuseõpe, eriala: Informaatika ja inseneriteadus.

Teenistuskäik

2024-	Cybernetica AS, teadur
2020–2024	Tartu Ülikool, infosüsteemide nooremteadur
2019–2020	Tartu Ülikool, andmeanalüütik

Teadustegevus

Peamised uurimisvaldkonnad:

- masinõpe
- protsessikaeve

DISSERTATIONES INFORMATICAE PREVIOUSLY PUBLISHED IN DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS

19. **Helger Lipmaa.** Secure and efficient time-stamping systems. Tartu, 1999, 56 p.
22. **Kaili Müürisep.** Eesti keele arvutigrammatika: süntaks. Tartu, 2000, 107 lk.
23. **Varmo Vene.** Categorical programming with inductive and coinductive types. Tartu, 2000, 116 p.
24. **Olga Sokratova.** Ω-rings, their flat and projective acts with some applications. Tartu, 2000, 120 p.
27. **Tiina Puolakainen.** Eesti keele arvutigrammatika: morfoloogiline ühestamine. Tartu, 2001, 138 lk.
29. **Jan Villemson.** Size-efficient interval time stamps. Tartu, 2002, 82 p.
45. **Kristo Heero.** Path planning and learning strategies for mobile robots in dynamic partially unknown environments. Tartu 2006, 123 p.
49. **Härmel Nestra.** Iteratively defined transfinite trace semantics and program slicing with respect to them. Tartu 2006, 116 p.
53. **Marina Issakova.** Solving of linear equations, linear inequalities and systems of linear equations in interactive learning environment. Tartu 2007, 170 p.
55. **Kaarel Kaljurand.** Attempto controlled English as a Semantic Web language. Tartu 2007, 162 p.
56. **Mart Anton.** Mechanical modeling of IPMC actuators at large deformations. Tartu 2008, 123 p.
59. **Reimo Palm.** Numerical Comparison of Regularization Algorithms for Solving Ill-Posed Problems. Tartu 2010, 105 p.
61. **Jüri Reimand.** Functional analysis of gene lists, networks and regulatory systems. Tartu 2010, 153 p.
62. **Ahti Peder.** Superpositional Graphs and Finding the Description of Structure by Counting Method. Tartu 2010, 87 p.
64. **Vesal Vojdani.** Static Data Race Analysis of Heap-Manipulating C Programs. Tartu 2010, 137 p.
66. **Mark Fišel.** Optimizing Statistical Machine Translation via Input Modification. Tartu 2011, 104 p.
67. **Margus Niitsoo**. Black-box Oracle Separation Techniques with Applications in Time-stamping. Tartu 2011, 174 p.
71. **Siim Karus.** Maintainability of XML Transformations. Tartu 2011, 142 p.
72. **Margus Treumuth.** A Framework for Asynchronous Dialogue Systems: Concepts, Issues and Design Aspects. Tartu 2011, 95 p.
73. **Dmitri Lepp.** Solving simplification problems in the domain of exponents, monomials and polynomials in interactive learning environment T-algebra. Tartu 2011, 202 p.

74. **Meelis Kull.** Statistical enrichment analysis in algorithms for studying gene regulation. Tartu 2011, 151 p.
77. **Bingsheng Zhang.** Efficient cryptographic protocols for secure and private remote databases. Tartu 2011, 206 p.
78. **Reina Uba.** Merging business process models. Tartu 2011, 166 p.
79. **Uuno Puus.** Structural performance as a success factor in software development projects – Estonian experience. Tartu 2012, 106 p.
81. **Georg Singer.** Web search engines and complex information needs. Tartu 2012, 218 p.
83. **Dan Bogdanov.** Sharemind: programmable secure computations with practical applications. Tartu 2013, 191 p.
84. **Jevgeni Kabanov.** Towards a more productive Java EE ecosystem. Tartu 2013, 151 p.
87. **Margus Freudenthal.** Simpl: A toolkit for Domain-Specific Language development in enterprise information systems. Tartu, 2013, 151 p.
90. **Raivo Kolde.** Methods for re-using public gene expression data. Tartu, 2014, 121 p.
91. **Vladimir Šor.** Statistical Approach for Memory Leak Detection in Java Applications. Tartu, 2014, 155 p.
92. **Naved Ahmed.** Deriving Security Requirements from Business Process Models. Tartu, 2014, 171 p.
94. **Liina Kamm.** Privacy-preserving statistical analysis using secure multiparty computation. Tartu, 2015, 201 p.
100. **Abel Armas Cervantes.** Diagnosing Behavioral Differences between Business Process Models. Tartu, 2015, 193 p.
101. **Fredrik Milani.** On Sub-Processes, Process Variation and their Interplay: An Integrated Divide-and-Conquer Method for Modeling Business Processes with Variation. Tartu, 2015, 164 p.
102. **Huber Raul Flores Macario.** Service-Oriented and Evidence-aware Mobile Cloud Computing. Tartu, 2015, 163 p.
103. **Tauno Metsalu.** Statistical analysis of multivariate data in bioinformatics. Tartu, 2016, 197 p.
104. **Riivo Talviste.** Applying Secure Multi-party Computation in Practice. Tartu, 2016, 144 p.
108. **Siim Orasmaa.** Explorations of the Problem of Broad-coverage and General Domain Event Analysis: The Estonian Experience. Tartu, 2016, 186 p.
109. **Prastudy Mungkas Fauzi.** Efficient Non-interactive Zero-knowledge Protocols in the CRS Model. Tartu, 2017, 193 p.
110. **Pelle Jakovits.** Adapting Scientific Computing Algorithms to Distributed Computing Frameworks. Tartu, 2017, 168 p.
111. **Anna Leontjeva.** Using Generative Models to Combine Static and Sequential Features for Classification. Tartu, 2017, 167 p.
112. **Mozhgan Pourmoradnasseri.** Some Problems Related to Extensions of Polytopes. Tartu, 2017, 168 p.

113. **Jaak Randmets.** Programming Languages for Secure Multi-party Computation Application Development. Tartu, 2017, 172 p.
114. **Alisa Pankova.** Efficient Multiparty Computation Secure against Covert and Active Adversaries. Tartu, 2017, 316 p.
116. **Toomas Saarsen.** On the Structure and Use of Process Models and Their Interplay. Tartu, 2017, 123 p.
121. **Kristjan Korjus.** Analyzing EEG Data and Improving Data Partitioning for Machine Learning Algorithms. Tartu, 2017, 106 p.
122. **Eno Tõnisson.** Differences between Expected Answers and the Answers Offered by Computer Algebra Systems to School Mathematics Equations. Tartu, 2017, 195 p.

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS

1. **Abdullah Makkeh**. Applications of Optimization in Some Complex Systems. Tartu 2018, 179 p.
2. **Riivo Kikas**. Analysis of Issue and Dependency Management in Open-Source Software Projects. Tartu 2018, 115 p.
3. **Ehsan Ebrahimi**. Post-Quantum Security in the Presence of Superposition Queries. Tartu 2018, 200 p.
4. **Ilya Verenich**. Explainable Predictive Monitoring of Temporal Measures of Business Processes. Tartu 2019, 151 p.
5. **Yauhen Yakimenka**. Failure Structures of Message-Passing Algorithms in Erasure Decoding and Compressed Sensing. Tartu 2019, 134 p.
6. **Irene Teinemaa**. Predictive and Prescriptive Monitoring of Business Process Outcomes. Tartu 2019, 196 p.
7. **Mohan Liyanage.** A Framework for Mobile Web of Things. Tartu 2019, 131 p.
8. **Toomas Krips.** Improving performance of secure real-number operations. Tartu 2019, 146 p.
9. **Vijayachitra Modhukur.** Profiling of DNA methylation patterns as biomarkers of human disease. Tartu 2019, 134 p.
10. **Elena Sügis.** Integration Methods for Heterogeneous Biological Data. Tartu 2019, 250 p.
11. **Tõnis Tasa.** Bioinformatics Approaches in Personalised Pharmacotherapy. Tartu 2019, 150 p.
12. **Sulev Reisberg.** Developing Computational Solutions for Personalized Medicine. Tartu 2019, 126 p.
13. **Huishi Yin.** Using a Kano-like Model to Facilitate Open Innovation in Requirements Engineering. Tartu 2019, 129 p.
14. **Faiz Ali Shah.** Extracting Information from App Reviews to Facilitate Software Development Activities. Tartu 2020, 149 p.
15. **Adriano Augusto**. Accurate and Efficient Discovery of Process Models from Event Logs. Tartu 2020, 194 p.
16. **Karim Baghery.** Reducing Trust and Improving Security in zk-SNARKs and Commitments. Tartu 2020, 245 p.
17. **Behzad Abdolmaleki.** On Succinct Non-Interactive Zero-Knowledge Protocols Under Weaker Trust Assumptions. Tartu 2020, 209 p.
18. **Janno Siim.** Non-Interactive Shuffle Arguments. Tartu 2020, 154 p.
19. **Ilya Kuzovkin.** Understanding Information Processing in Human Brain by Interpreting Machine Learning Models. Tartu 2020, 149 p.
20. **Orlenys López Pintado.** Collaborative Business Process Execution on the Blockchain: The Caterpillar System. Tartu 2020, 170 p.
21. **Ardi Tampuu.** Neural Networks for Analyzing Biological Data. Tartu 2020, 152 p.

22. **Madis Vasser.** Testing a Computational Theory of Brain Functioning with Virtual Reality. Tartu 2020, 106 p.
23. **Ljubov Jaanuska.** Haar Wavelet Method for Vibration Analysis of Beams and Parameter Quantification. Tartu 2021, 192 p.
24. **Arnis Parsovs.** Estonian Electronic Identity Card and its Security Challenges. Tartu 2021, 214 p.
25. **Kaido Lepik.** Inferring causality between transcriptome and complex traits. Tartu 2021, 224 p.
26. **Tauno Palts.** A Model for Assessing Computational Thinking Skills. Tartu 2021, 134 p.
27. **Liis Kolberg.** Developing and applying bioinformatics tools for gene expression data interpretation. Tartu 2021, 195 p.
28. **Dmytro Fishman.** Developing a data analysis pipeline for automated protein profiling in immunology. Tartu 2021, 155 p.
29. **Ivo Kubjas.** Algebraic Approaches to Problems Arising in Decentralized Systems. Tartu 2021, 120 p.
30. **Hina Anwar.** Towards Greener Software Engineering Using Software Analytics. Tartu 2021, 186 p.
31. **Veronika Plotnikova.** FIN-DM: A Data Mining Process for the Financial Services. Tartu 2021, 197 p.
32. **Manuel Camargo.** Automated Discovery of Business Process Simulation Models From Event Logs: A Hybrid Process Mining and Deep Learning Approach. Tartu 2021, 130 p.
33. **Volodymyr Leno.** Robotic Process Mining: Accelerating the Adoption of Robotic Process Automation. Tartu 2021, 119 p.
34. **Kristjan Krips.** Privacy and Coercion-Resistance in Voting. Tartu 2022, 173 p.
35. **Elizaveta Yankovskaya.** Quality Estimation through Attention. Tartu 2022, 115 p.
36. **Mubashar Iqbal.** Reference Framework for Managing Security Risks Using Blockchain. Tartu 2022, 203 p.
37. **Jakob Mass.** Process Management for Internet of Mobile Things. Tartu 2022, 151 p.
38. **Gamal Elkoumy.** Privacy-Enhancing Technologies for Business Process Mining. Tartu 2022, 135 p.
39. **Lidia Feklistova.** Learners of an Introductory Programming MOOC: Background Variables, Engagement Patterns and Performance. Tartu 2022, 151 p.
40. **Mohamed Ragab.** Bench-Ranking: A Prescriptive Analysis Approach for Large Knowledge Graphs Query Workloads. Tartu 2022, 158 p.
41. **Mohammad Anagreh.** Privacy-Preserving Parallel Computations for Graph Problems. Tartu 2023, 181 p.
42. **Rahul Goel.** Mining Social Well-being Using Mobile Data. Tartu 2023, 104 p.

43. **Anti Ingel.** Algorithms using information theory: classification in brain-computer interfaces and characterising reinforcement-learning agents. Tartu 2023, 142 p.
44. **Shakshi Sharma.** Fighting Misinformation in the Digital Age: A Comprehensive Strategy for Characterizing, Identifying, and Mitigating Misinformation on Online Social Media Platforms. Tartu 2023, 158 p.
45. **Kristiina Rahkema.** Quality Analysis of iOS Applications with Focus on Maintainability and Security Aspects. Tartu 2023, 182 p.
46. **Ivan Slobozhan.** Studying Online Social Media Engagement in CIS Countries during Protests, Mass Demonstrations and War. Tartu 2023, 81 p.
47. **Nurlan Kerimov.** Building a catalogue of molecular quantitative trait loci to interpret complex trait associations. Tartu 2023, 248 p.
48. **Pavlo Tertychnyi.** Machine Learning Methods for Anti-Money Laundering Monitoring. Tartu 2023, 117 p.
49. **Abasi-amefon Obot Affia.** A Framework and Teaching Approach for IoT Security Risk Management. Tartu 2023, 180 p.
50. **Raimond-Hendrik Tunnel.** Video Game Design and Development Bachelor's Curriculum for Estonia. Tartu 2024, 137 p.
51. **Ahto Salumets.** Bioinformatics analysis of various aspects in immunology. Tartu 2024, 198 p.
52. **Mohammed Abdulhameed Shaif Ali.** Deep Learning Methods for Cell Microscopy Image Analysis. Tartu 2024, 143 p.
53. **Pille Pullonen-Raudvere**. Foundations of Efficient and Secure Algorithm Development for Secure Multiparty Computation. Tartu 2024, 265 p.
54. **Marili Rõõm.** Multiple approaches to learners' success and factors affecting it in computer programming MOOCs. Tartu 2024, 170 p.
55. **Shivananda Rangappa Poojara.** Design and Orchestration of Scalable, Event-Driven Serverless Data Pipelines for Internet of Things (IoT) Applications. Tartu 2024, 172 p.
56. **Hassan Abdulgaleel Hassan Salim Eldeeb.** Empowering Machine Learning Pipelines with Automated Feature Engineering. Tartu 2024, 121 p.
57. **Muhammad Uzair.** Soft decision making for agri-food 4.0. Tartu 2024, 158 p.
58. **Kirill Milintsevich.** Estimation of Depression Level from Text: Symptom-Based Approach, External Knowledge, Dataset Validity. Tartu 2024, 130 p.
59. **Maksym Del.** Multilingual and Multi-Domain Representational Patterns Across Trpansformer-Based Models. Tartu 2024, 131 p.
60. **Kristo Raun.** Adaptive Out-of-order Handling in Streaming Conformance Checking. Tartu 2024, 118 p.
61. **Toivo Vajakas.** Towards integration of mobile network data into analyzing human mobility. Tartu 2024, 103 p.
62. **Katsiaryna Lashkevich.** Data-Driven Analysis and Optimization of Waiting Times in Business Processes. Tartu 2024, 169 p.
63. **Alejandra Duque-Torres.** Classifying, Constraining and Ranking Metamorphic Relations. Tartu 2025, 159 p.

64. **Mariia Bakhtina.** A Method for Information Security and Privacy Management in Smart Solutions. Tartu 2025, 199 p.
65. **Andre Tättar.** Multilingual Machine Translation for Under-Resourced Languages. Tartu 2025, 170 p.