Discrete Event Dynamic Systems (2025) 35:205–231
https://doi.org/10.1007/s10626-025-00419-4

Probabilistic timed Petri nets for clinical pathway design and analysis: a case study

Manon Le Moigne[1] · Cristian Mahulea[2] · Gregory Faraut[1] · Simona Bernardi[2] · Jorge Albareda[3] · Lidia Castán[3]

Received: 30 November 2024 / Accepted: 10 July 2025 / Published online: 23 July 2025

Abstract
The COVID-19 pandemic underscored the need for efficient hospital resource management and standardized patient care. Clinical pathways, structured plans for managing specific conditions, are critical but challenging to design and update. This paper introduces a methodology combining pattern mining and Probabilistic Timed Petri Nets (PTPN) to model, simulate, and evaluate clinical pathways, incorporating probabilistic transitions and continuous-time distributions for activity durations. In collaboration with medical professionals, the proposed methodology has been successfully applied to develop a clinical pathway for anterior cruciate ligament (ACL) ruptures using real data. Iterative refinement addressed data inconsistencies, producing a robust PTPN model that optimizes patient care and resource allocation. Validation using the CASAS database demonstrate the adaptability of the method to different healthcare contexts.

Keywords Probabilistic timed petri nets · Identification · Healthcare system · Clinical pathway · Process mining

1 Introduction

During the past two decades, advances in medical science, the development of new surgical techniques, and the increasing demand for healthcare —exacerbated by the COVID-19 pandemic— have forced hospitals to modernize their protocols and clinical pathways to improve efficiency, optimize resources, and improve patient satisfaction. Healthcare systems around the world face significant pressures in areas such as bed allocation, staffing, and patient care coordination. Clinical pathways, which are structured multidisciplinary plans for the management of specific conditions, are essential to optimize resources and standardize

This work was partially supported by Grant TED2021-130449B-I00 funded by MCIN/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR" and by Grants DAMOCLES-PID2020-113969RB-I00 and iSuma-PID2021-125514NB-I00 funded by the Ministerio de Ciencia e Innovación.

Extended author information available on the last page of the article patient care. However, the design and ongoing updating of these pathways require substantial resources and frequent adaptation to align with the changing demographics of patients, medical practices, and institutional resources (Vanhaecht et al. 2010; Mould et al. 2010).

Inconsistent treatment approaches among healthcare providers can lead to varying use of resources and outcomes for patients with the same condition. Clinical pathways address this by standardizing diagnostic and treatment resources, ensuring that all patients receive consistent care based on best practices. This standardization promotes quality care, minimizes unnecessary resource use, and leads to more homogeneous patient outcomes.

Traditional approaches to developing clinical pathways are often manual and rely heavily on consensus among healthcare teams. Although effective, these methods are labor-intensive and may not be efficient in high-demand situations, such as during pandemics or resource shortages (Hindle et al. 2006). Automated methods for generating and updating clinical pathways can address these challenges. This paper introduces a formal methodology using Probabilistic Timed Petri Nets (PTPN) to model, analyze, and optimize clinical pathways. By incorporating probabilistic and temporal elements, PTPNs provide a structured yet flexible representation of patient care flows, effectively modeling the variability inherent in healthcare processes.

Building on previous work (Le Moigne et al. 2024), this study makes four main contributions:

- **Integration of continuous-time distributions**: By incorporating continuous-time distributions to represent activity durations, this study offers a more accurate depiction of real-world variability in patient care. Although earlier models used time intervals, this approach enhances performance evaluations, such as throughput estimation (Bernardi et al. 2019; Fouquet et al. 2020).
- **Comprehensive methodology implementation**: The implementation, in Python, of all algorithms in this paper is open-source, available at Gitlab: https://gitlab.crans.org/bleizi/probabilistic-time-petri-nets-for-clinical-pathway-design-and-analysis. This includes a PTPN module that allows users to simulate and evaluate PTPN models in various scenarios, providing a practical tool to analyze and optimize clinical pathways.
- **Computation of performance bounds**: Another contribution is the framework for computing performance bounds (e.g., throughput, cycle time) based on the translation of clinical pathways into Stochastic Petri Net (SPN) models. This allows for a detailed and stochastic performance analysis of the pathways.
- **Expanded case study and validation**: Using real-world data from the Orthopedic Surgery and Traumatology Department at the Hospital Clínico Universitario (HCU) in Zaragoza, this paper shows the adaptability of the model with an extensive case study on a clinical pathway for anterior cruciate ligament (ACL) ruptures. Validation using data from the CASAS project, which involves daily living activities, further illustrates the adaptability of the model, enabling more precise performance evaluations and resource optimization.

The implementation of a clinical pathway provides *multiple advantages* both for patients and the healthcare system. These key benefits include:

- **Health benefits**: Clinical pathways ensure consistent and evidence-based treatment for all patients, leading to standardized outcomes. This alignment allows medical professionals to apply a uniform approach in treating patients with the same condition.

- **Economic benefits**: Using only the resources necessary to treat a given condition, clinical pathways promote cost-effective healthcare management. This approach not only optimizes resource use, but also minimizes the potential risks to patients associated with unnecessary treatments.
- **Legal benefits**: Clinical pathways help ensure that all essential resources are used to treat each condition while avoiding unnecessary interventions. This adherence to standards improves legal and ethical compliance, promoting safer and more standardized clinical practices.

This paper addresses the limitations of traditional process mining tools, such as ProM (Aalst et al. 2009) and PM4PY (Berti et al. 2023), which focus primarily on concurrency and may struggle with the inherently *sequential nature of clinical pathways* (van der Aalst 2016; Rifki et al. 2024; van der Aalst and Carmona 2022). In contrast, the proposed pattern discovery algorithm emphasizes sequential events, directly identifying frequent patterns and associating them with probabilistic transitions. This approach provides a comprehensive view of patient care that incorporates both timing and variability, leading to improved information on treatment sequences (Emzivat et al. 2016).

PTPNs provide a rigorous yet intuitive framework for capturing the inherent uncertainty and timing constraints of clinical workflows. Building on the foundation of Time Petri Nets (TPNs), they introduce probabilistic post-arcs that cleanly separate “when” an activity can occur from “which” outcome will follow, a distinction that is crucial in medicine, where both the duration of an intervention (e.g., drug infusion time) and the likelihood of different patient responses must be modeled. By equipping each transition with a continuous firing-time distribution and each outgoing arc with a probability, PTPNs deliver a compact, transparent representation of real-world care pathways. This dual timing–probability structure not only supports high-fidelity simulation of individual patient journeys, but also enables “what-if” analyses of resource allocations and branching treatment strategies. In contrast to earlier work (Emzivat et al. 2016), which uses time intervals on transitions, here we fit full continuous distributions, e.g., log-normal, gamma, or exponential, to observed activity durations, allowing us to derive quantitative performance measures (throughput, cycle time) directly from the model.

The proposed method bridges the gap between raw clinical data and actionable insights for healthcare professionals. By simulating pathways under different scenarios, this approach improves resource planning and patient flow management. The model is also scalable and adaptable, making it suitable for real-time hospital management, particularly during periods of high demand, such as pandemics.

The remainder of this paper is organized as follows. Section 2 introduces the proposed methodology, while Section 3 details the pattern discovery algorithm. Section 4 describes the PTPN model and Section 5 describes the framework for the computation of performance bounds in PTPN models. In Section 6, we apply the proposed methodology in a real-world case study to develop a clinical pathway. Finally, Section 7 provides conclusions and discusses future research directions.

2 Overview of the proposed method

The methodology proposed in this paper aims to develop a formal model of clinical pathways that takes advantage of hospital-recorded events. This model is designed to facilitate comprehensive analysis, improve patient monitoring, and assist medical professionals in optimizing clinical pathways. The structure of the method is shown in Fig. 1.

The foundation of our method is a unified event log in which all activities of the patient, covering the full course of a given pathology, are recorded as an ordered sequence. By organizing the raw data into patient traces, we keep the analysis focused on individual care journeys. These traces then feed into process-mining routines that automatically infer a formal model whose behavior reproduces all observed sequences. In prior work (Le Moigne et al. 2024) we evaluated three mining strategies: (1) *inductive algorithm* (an adaptation of the Alpha algorithm (van der Aalst 2010)); (2) *heuristic algorithm* (based on frequency-and-dependency heuristics (Weijters et al. 2006)) and (3) *pattern-discovery algorithm* (extracting frequent sequential subsequences (Saives and Faraut 2014; Saives et al. 2015)).

As we show in Le Moigne et al. (2024), the pattern-discovery approach delivers the most faithful representation of clinical pathways, because it preserves the strictly sequential nature of treatment steps rather than introducing *spurious concurrency*. A schematic of this pipeline appears at the top of Fig. 1.

Therefore, in this paper first, we apply the pattern-discovery algorithm to the patient event logs, extracting the most frequent two-step sequences and their support counts. Next, we feed those mined patterns into our PTPN construction library (Section 4), which automatically assembles places, transitions, probabilistic post-arcs and timing distributions. This Python toolkit can both simulate the resulting net and generate graphical renderings of the PTPN for easy inspection. The visual output helps medical teams see where care pathways branch or bottleneck and suggests opportunities for streamlining. After that, each discovered pattern is post-processed into a discrete probability distribution over its

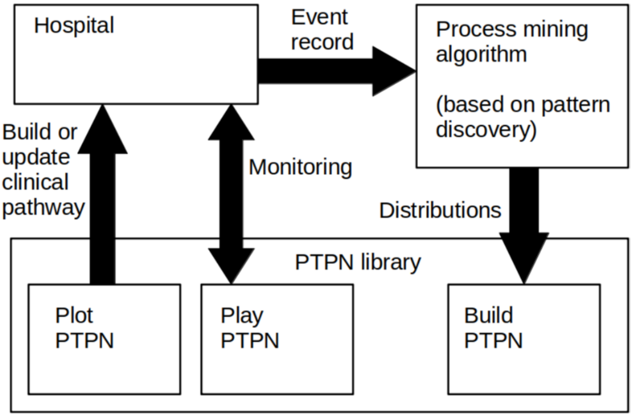


Fig. 1 Overview of the proposed method

possible successors; those probabilities label the post-arcs of the PTPN, ensuring the net faithfully reproduces the statistics observed in the original data.

The initial PTPN model is then refined in collaboration with medical personnel. This validation step is essential because the data used to construct the model may be imperfect; for example, some events may have been missed for various reasons. This multidisciplinary task, which involves both engineers and medical professionals, ensures the precision and clinical relevance of the pathway.

Once the PTPN model is built and validated, it can be used to update the clinical pathway. The Python library allows for exporting a graphical representation of the pathway that can be shared with medical personnel. In addition, PTPN can be "played" to monitor patient progress if PTPN and hospital processes evolve simultaneously.

If clinical pathways for multiple pathologies are obtained, resource usage and time delays associated with transitions can also be analyzed. This model can then be used to evaluate the performance of the system under different conditions. For example, the impact of increasing the number of doctors on patient flow can be studied, making this model a useful tool for healthcare management, as proposed in Bernardi et al. (2019); Mahulea et al. (2018).

3 Discovering patterns

A process mining method can be used to extract data from a database to obtain a formal model. Our database is composed of the sequence of events applied to each patient. As mentioned above, three algorithms were tested and needed to produce a formal model, with results that are coherent with a clinical pathway according to medical doctors: inductive algorithm, heuristic algorithm, and pattern discovery. The algorithm, developed in this paper, is an adapted version of the pattern discovery algorithm proposed in Saives and Faraut (2014); Saives et al. (2015).

The methodology for extracting clinical pathways from event logs relies on identifying recurring patterns of events that represent the sequence of activities performed during the treatment process. This section details the database structure, the pattern discovery algorithm, and the process of transforming these patterns into probabilistic models for further analysis.

The *database* used in this study consists of event logs, where each log captures the sequence of activities performed for an individual patient during its clinical pathway. Each event log in the database contains the following information.

- *Patient ID*. An anonymized identifier for the patient.
- *Sequence of events*. An ordered list of events (starting of activities) represented by unique labels ($e_1, e_2, \dots$). These events are specific to the clinical pathway being analyzed.
- *Timestamps (optional)*. A timestamp can be associated to each event of the sequence. This field is optional, since timestamps are not always available.

A *pattern* is a recurring sequence of two events, observed in multiple patient event logs. A pattern is represented as an ordered pair of $\langle e_i \rightarrow e_j \rangle$, where e_i and e_j are events in the database. Pattern discovery consists in detecting patterns in the database of patient logs and determining their *support*, i.e., the number of occurrences of the pattern in the patient logs.

Algorithm 1 Pattern discovery algorithm with 95% Wilson confidence intervals.

```
Require: Database of patient logs (each a sequence of event labels)
Ensure: list_disc = list of tuples $(\langle e_i \rightarrow e_j \rangle, k_{ij}, n_i, \pi_{ij}, \mathrm{CI}_{95\%})$
 1: $\mathcal{E}$ := set of all event labels in database
 2: Initialize count[ $\langle e_i, e_j \rangle$ ] := 0 for every ordered pair in $\mathcal{E} \times \mathcal{E}$
 3: Initialize total[ $e$ ] := 0 for every $e \in \mathcal{E}$    ▷ Count occurrences of each length-2 pattern and of each
    precursor
 4: for all sequence $S$ in database do
 5:     for $r = 1$ to $|S| - 1$ do
 6:         $e_i := S[r],\ e_j := S[r+1]$
 7:         count[$\langle e_i, e_j \rangle$] += 1
 8:         total[$e_i$ ] += 1
 9:     end for
10: end for
11: $z \leftarrow 1.96$                                          ▷ z-value for 95% confidence
12: list_disc ← [ ]
13: for all $\langle e_i, e_j \rangle$ with $n_i$ := total[$e_i$] > 0 do
14:     $k_{ij}$ := count[$\langle e_i, e_j \rangle$]
15:     $\pi_{ij} := k_{ij}/n_i$
16:     $\tilde{\pi}_{ij} := \frac{\pi_{ij} + \frac{z^2}{2n_i}}{1+\frac{z^2}{n_i}}$
17:     half := $\frac{z}{1+\frac{z^2}{n_i}} \sqrt{\frac{\pi_{ij}(1-\pi_{ij})}{n_i} + \frac{z^2}{4n_i^2}}$
18:     $\mathrm{CI}_{95\%} := [\, \tilde{\pi}_{ij} - \mathrm{half},\ \tilde{\pi}_{ij} + \mathrm{half} \,]$
19:     Append $(\langle e_i \rightarrow e_j \rangle, k_{ij}, n_i, \pi_{ij}, \mathrm{CI}_{95\%})$ to list_disc
20: end for
21: return list_disc
```

Alg. 1 proceeds in three phases. First, it enumerates every ordered pair of event labels $\langle e_i, e_j \rangle$ (step 1) and counts two quantities: (a) k_{ij}, the number of times e_j immediately follows e_i in any patient trace, and (b) n_i, the total number of times e_i appears as a precursor (loop in steps 4-10). Second, for each pair with $n_i > 0$, it computes the empirical transition probability

$$\pi_{ij} \ = \ \frac{k_{ij}}{n_i}$$

and then constructs a 95% Wilson confidence interval $\mathrm{CI}_{95\%}([\pi_{ij}])$ around that estimate. This interval accounts for sampling variation; very wide intervals highlight low-support patterns that merit expert review; while always remaining within [0, 1].

Finally, the algorithm outputs $list\_disc$, a list of tuples

$$(\langle e_i \rightarrow e_j \rangle,\ k_{ij},\ n_i,\ \pi_{ij},\ \mathrm{CI}_{95\%})$$

for every pattern with $k_{ij} > 0$. These entries succinctly summarize both the point estimate and the statistical uncertainty of each transition probability, forming a robust basis for building the PTPN.

Example 1 Consider, as a small example, a database of three logs with three patients undergoing anterior cruciate ligament (ACL) surgery. We focus on the first day after surgery and the goal is to identify recurring patterns of activities in the patients' treatment process.

The events presented in the logs correspond to the starting of a set of seven activities, namely,

- e_1 - Drain removal
- e_2 - Checking the surgical wound
- e_3 - Stained dressing
- e_4 - Intrahospital_AB_prolongation
- e_5 - Wandering
- e_6 - Analgesic rescue
- e_7 - Hospital discharge

A small part of the database that will be used in Section 6 containing only three logs for three patients is considered in this section, namely,

- *Patient 1*: $(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$,
- *Patient 2*: $(e_1, e_2, e_5, e_6, e_7)$,
- *Patient 3*: $(e_1, e_2, e_3, e_4, e_5, e_7)$.

The part of the considered database consists of sequences of activities for these patients, recovering from anterior cruciate ligament (ACL) surgery following standard post-operative procedures, with specific variations based on individual needs.

- *Patient 1* begins with drain removal (e_1), followed by check of the surgical wound (e_2), which reveals a stained dressing (e_3). This leads to a decision to prolong antibiotics (Intrahospital_AB_prolongation - e_4) to prevent infection. The patient is then encouraged to start light physical activity (wandering - e_5) and requires additional pain relief (analgesic rescue - e_6) before discharge (e_7).
- For *Patient 2*, the sequence involves drain removal (e_1) and wound check (e_2), followed by light movement (wandering - e_5) and the administration of additional pain relief (e_6) before discharge (e_7).
- In the case of *Patient 3*, after the drain is removed (e_1) and the wound is checked (e_2), a stained dressing is observed (e_3), leading to prolonged antibiotic use (e_4). The patient then engages in light physical activity (wandering - e_5) before being discharged (e_7).

These sequences illustrate standard postoperative care, while also highlighting individualized responses such as extended antibiotic use or additional analgesics when necessary.

Applying Alg. 1 we first count for each ordered pair $\langle e_i \rightarrow e_j \rangle$:

$$n_i = \text{total occurrences of } e_i, \quad k_{ij} = \text{times } e_j \text{ follows } e_i.$$

Here:

e_i	e_1	e_2	e_3	e_4	e_5	e_6	e_7
n_i	3	3	2	2	3	2	0

$\langle e_i \rightarrow e_j \rangle$	k_{ij}
$\langle e_1 \rightarrow e_2 \rangle$	3
$\langle e_2 \rightarrow e_3 \rangle$	2
$\langle e_2 \rightarrow e_5 \rangle$	1
$\langle e_3 \rightarrow e_4 \rangle$	2
$\langle e_4 \rightarrow e_5 \rangle$	2
$\langle e_5 \rightarrow e_6 \rangle$	2
$\langle e_5 \rightarrow e_7 \rangle$	1
$\langle e_6 \rightarrow e_7 \rangle$	2

Next step is to estimate $\pi_{ij} = k_{ij}/n_i$ and their 95% Wilson intervals $[\ell_{ij}, u_{ij}]$:

$\langle e_i \rightarrow e_j \rangle$	π_{ij}	k_{ij}	n_i	$[\ell_{ij}, u_{ij}]_{95\%}$
$\langle e_1 \rightarrow e_2 \rangle$	1.000	3	3	[0.44, 1.00]
$\langle e_2 \rightarrow e_3 \rangle$	0.667	2	3	[0.21, 0.94]
$\langle e_2 \rightarrow e_5 \rangle$	0.333	1	3	[0.06, 0.79]
$\langle e_3 \rightarrow e_4 \rangle$	1.000	2	2	[0.34, 1.00]
$\langle e_4 \rightarrow e_5 \rangle$	1.000	2	2	[0.34, 1.00]
$\langle e_5 \rightarrow e_6 \rangle$	0.667	2	3	[0.21, 0.94]
$\langle e_5 \rightarrow e_7 \rangle$	0.333	1	3	[0.06, 0.79]
$\langle e_6 \rightarrow e_7 \rangle$	1.000	2	2	[0.34, 1.00]

Thus $list\_disc$ becomes

$$(\langle e_i \rightarrow e_j \rangle,\ k_{ij},\ n_i,\ \pi_{ij},\ [\ell_{ij}, u_{ij}]) \quad \text{for each } k_{ij} > 0.$$

For example, for e_2 we have $n_2 = 3$, two successors e_3 ($k_{23} = 2$) and e_5 ($k_{25} = 1$), yielding

$$\pi_{23} = 2/3,\ [0.21, 0.94], \quad \pi_{25} = 1/3,\ [0.06, 0.78].$$

□

For the time complexity analysis of Alg. 1, let *n* denote the number of patients (traces) in the database, $m = \max_{S \in \text{database}} |S|$ denote the length of the longest trace, and $\ell = |\mathcal{E}|$ represent the size of the event alphabet. The complexity is analyzed as follows:

1. **Building the alphabet (scanning all traces, line 1).** The complexity depends on the data structure chosen for storing the alphabet. If a balanced tree or a sorted set is used, each insertion (including the existence check) takes $O(\log \ell)$ time. With *n* traces and at most *m* events per trace, the worst-case complexity is thus $O(nm \log \ell)$.

In the degenerate scenario where each event is unique, implying $\ell = nm$, the complexity is $O(nm\log(nm))$. Alternatively, an unsorted list structure would yield a linear search for each event insertion, resulting in $O(nm\ell)$ complexity, which in the worst case ($\ell = nm$) is $O(n^2m^2)$. Using a hash-based set, which typically provides constant-time insertion and lookup on average, would result in an average-case complexity of $O(nm)$.

2. **Initializing the count and total tables (lines 2 and 3).** This step involves initializing counts for all ordered event pairs from $\mathcal{E} \times \mathcal{E}$, resulting in a complexity of $O(\ell^2)$.
3. **Counting adjacent pairs (loop in lines 4–10).** The nested loops iterate over each trace once, counting adjacent event pairs. This operation clearly has complexity $O(nm)$.
4. **Computing probabilities and Wilson intervals (loop in lines 13–20).** This final loop processes each of the up to ℓ^2 possible event pairs individually, each in constant time, thus requiring $O(\ell^2)$.

Summing these complexities, the total complexity becomes:

$$O(nm\log\ell + \ell^2 + nm + \ell^2) = O(nm\log\ell + \ell^2).$$

In the degenerate worst case scenario ($\ell = O(nm)$), the overall complexity is:

$$O(nm\log(nm) + (nm)^2) = O((nm)^2).$$

In any case, the algorithm remains polynomial in the input size.

4 Building of a PTPN

To capture a clinical pathway in a way that is both analyzable and actionable, our formal model must satisfy four essential requirements. First, it must have *rigorous semantics*, i.e., be a fully specified mathematical object amenable to reachability analysis, simulation, and performance bounding. Second, it must offer *clinical readability*, so that medical staff can read, validate, and even adjust the model without deep formal methods training; its notation and graphical layout should feel familiar. Third, it must support *stochastic branching* to reflect observed variability in patient journeys: by attaching probabilities to alternative transitions, the net reproduces the same branching statistics seen in the data. Finally, it must provide *temporal fidelity*: each transition carries a full continuous-time distribution fit to observed activity durations, enabling us to predict not only which path a patient will follow but also how long each step takes, which is critical for resource planning and “what-if” timing analyses.

We chose a PN formalism because its graphical representation is immediately intelligible to clinicians, yet underpinned by a solid mathematical theory. In particular, the Probabilistic Timed Petri Net (PTPN) extends the familiar place/transition PNs with two orthogonal features: (a) probabilistic post-arcs and (b) continuous firing time distributions; so that it can simultaneously capture the branching uncertainty of patient decisions and the real-world timing of medical activities.

Definition 1 A Probabilistic Timed Petri net (PTPN) system is a tuple $\Sigma = \langle P, T, Pre, Post, D, \rho_{\mathcal{N}} \rangle$ where:

- P is a finite, non-empty set of places,
- T is a finite set of transitions such that $T \cap P = \emptyset$,
- $Pre : T \rightarrow \mathbb{N}^P$ is the backward incidence mapping,
- $Post : T \rightarrow \mathcal{P}\left(\mathcal{D}ist_{\mathbb{N}^P}\right)$ is the forward incidence mapping,
- $D : T \rightarrow \mathcal{D}(\mathbb{R}_+)$ is a function that assigns a firing continuous time distribution to each transition.
- $\rho_{\mathcal{N}} \in \mathcal{D}ist_{\mathbb{N}^P}$ is the distribution of initial marking of the net.

This definition is an extended version of the definition proposed in Emzivat et al. (2016), but instead of firing intervals, firing continuous time distributions are used (as exponential or log-normal laws). This modification allows for a more precise representation of event durations, capturing the inherent variability in the time required for different activities. This improves the model's ability to simulate and evaluate performance metrics.

The function $Post$ assigns to each transition a finite collection of discrete probability distributions over possible output-place markings. For any transition t, we treat the distributions in $Post(t)$ as driven by independent random variables: when t fires, one of these distributions is sampled to determine how many tokens appear in each successor place. We further assume each distribution in $Post(t)$ has finite support. Visually, a distribution in $Post(t)$ is depicted as a set of hyperarcs: each hyperarc carries a probability label, and when it is chosen, it unfolds into ordinary arcs that deposit the specified number of tokens into each of the output places. By using PTPN to model the patient flow in the healthcare system, only *one random variable* is needed to represent the forward incidence mapping. This approach models the advancement of a single patient in the clinical pathway, where one token is consumed and another is produced. Thus, $Post$ can also be represented by a matrix, $\text{Post} \in \mathbb{R}^{|P| \times |T|}$, with $\text{Post}[p_i, t_j]$ denoting the probability of producing a token in place p_i when firing t_j. Likewise, the function Pre can be represented by a matrix in our model, where all elements are either 0 or $+1$. Each transition has only one input place, reflecting the advancement of a single patient in the clinical pathway, and consequently the matrix $\text{Pre} \in \{0,1\}^{|P| \times |T|}$.

Table 1 Statistical distances between time distributions and measurements for the activity

Distribution	Energy distance	Wasserstein distance
log-normal	0.6	9
normal	1.3	21
exponential	5.6	103
gamma	0.5	8

Algorithm 2 From mined patterns to a PTPN.

```
Require: Database of patient logs (with time durations)
Require: list_disc                          ▷ list of tuples $(\langle e_i \to e_j \rangle, k, n, \pi, \mathrm{CI}_{95\%})$
Ensure: PTPN $\Sigma = \langle P, T, Pre, Post, D, \rho \rangle$
   /* Phase A: fit one continuous law per activity*/
 1: TimeDist $:= \emptyset$
 2: for all event label $e$ occurring in list_disc do
 3:     $\mathcal{T}_e :=$ observed durations of $e$
 4:     for each family $F \in$ {normal, exponential, log-normal, gamma} do
 5:         fit parameters $\theta_F$ on $\mathcal{T}_e$
 6:         compute $(E_F, W_F)$                ▷ energy & Wasserstein distances
 7:     end for
 8:     select $F^\star = \arg\min(E_F, W_F)$
 9:     TimeDist[$e$] $:= F^\star(\theta_{F^\star})$
10: end for
   /* Phase B: build places, transitions and arcs */
11: $P := \{start, end\}, \quad T := \emptyset$
12: for all label $e$ in TimeDist do
13:     create transition $t_e$ with $D(t_e) =$ TimeDist[$e$]
14:     $T := T \cup \{t_e\}$
15: end for
16: for all $(\langle e_i \to e_j \rangle, k, n, \pi, \mathrm{CI}_{95\%})$ in list_disc do
17:     ensure places $p_{e_i}, p_{e_j} \in P$
18:     add backward arc $p_{e_i} \to t_{e_i}$ of weight 1
19:     add probabilistic post-arc $t_{e_i} \xrightarrow{\pi} p_{e_j}$
20: end for
21: for all transition $t_e$ with ${}^\bullet t_e = \emptyset$ do
22:     add arc $start \to t_e$
23: end for
24: for all transition $t_e$ with $t_e^\bullet = \emptyset$ do
25:     add arc $t_e \to end$
26: end for
27: $\rho(start) = 1$ and $\rho(p) = 0, \forall p \in P \setminus \{start\}$.
28: return $\Sigma = \langle P, T, Pre, Post, D, \rho \rangle$
```

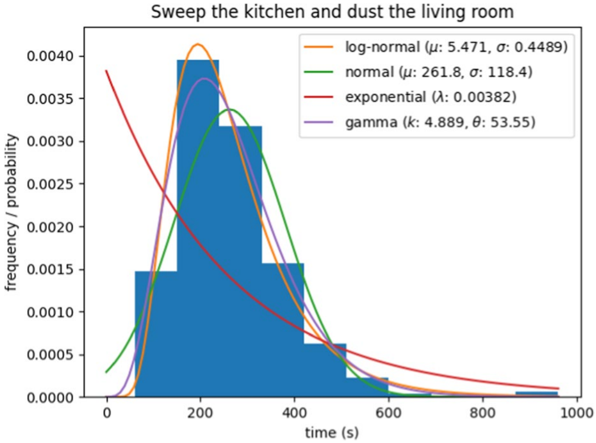
The construction of the PTPN relies upon the mined patterns produced by Alg. 1 and, when available, the duration information in the original dataset of event logs.

As summarized in Alg. 2, we proceed in two stages.

A. **Estimating timing laws**. For every event label e in the mined patterns, we collect the multiset of observed durations $\mathcal{T}_e$ and fit four candidate distributions—normal, exponential, log-normal and gamma—using maximum-likelihood estimation. We pick the law that minimizes the pair $(\text{energy distance}, \text{Wasserstein distance})$ between the fitted density and the empirical histogram (see e.g., Table 1 and Fig. 2). The best-fit continuous distribution is then assigned to the event (line 9).

B. **Getting the net structure**. Using the pattern list list_disc—which already contains, for every ordered pair $\langle e_i \to e_j \rangle$, its support and the conditional probability π_{ij}—we instantiate places, transitions, backward arcs (all of weight 1) and probabilistic post-arcs. Two dummy places start and end make the net well-formed even when an activity has no predecessors or successors in the log.

Fig. 2 Time distributions and frequency of an activity with 340 occurrences



Alg. 2 is the complete version. In case timestamps are not available, the first phase of Alg. 2 is skipped and just the *untimed* version of PTPN, that is the PTPN without the firing continuous-time distribution, is generated.

In Fouquet et al. (2020), time distributions were used to represent activity durations for Ambient Assisted Living systems. Drawing inspiration from this approach, the normal, exponential, log-normal, and gamma distributions were implemented, with functions that fit experimental durations. Each distribution is identified in our algorithm by a common function like Least Squares and/or Maximum Likelihood estimation. After computing the distributions of each transition, we determine which is the most accurate. Two comparison functions are employed to evaluate the quality of these fits: the Wasserstein distance and the energy distance.

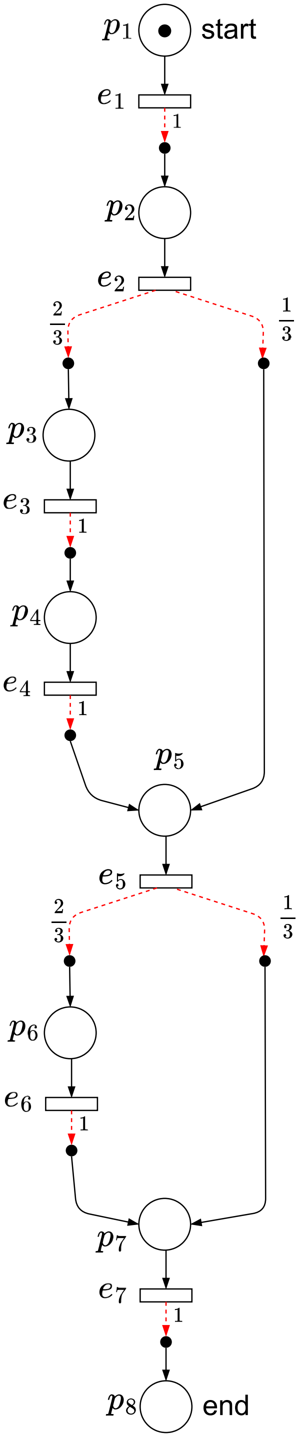
An example of a PTPN, derived from the discovered patterns and probabilities in Ex. 1, is shown in Fig. 3. The dotted red edges in the figure represent different possible outcomes from a probability distribution, with the respective probabilities indicated on the edges. This PTPN models the recovery process for patients undergoing ACL surgery and represents the sequence of events.

The model incorporates probabilistic branches. For example, after $Checking\_the\_surgical\_wound$ - transition labeled e_2, there is a 66.7% probability that the next activity will be $Stained\_dressing$ - place p_3 and a 33.3% chance that it will proceed to $Wandering$ - place p_5. Similarly, after $Wandering$ - transition e_5, the model reflects the likelihood that the patient will require $Analgesic\_rescue$ - place p_6 (66. 7%) or proceed directly to $Hospital\_discharge$ - place p_7 (33.3%).

5 PTPN performance bound computation

The enhancement of PTPN with continuous-time distributions, associated with transitions, enables evaluating performance metrics. In particular, we can resort to bounding techniques, proposed in the literature, for Time Petri Nets (TPN) (Bernardi and Campos 2013) and Stochastic Petri Nets (SPN) (Campos and Silva 1992) for this purpose. Bounding techniques are based on the derivation of a Linear Programming (LP) problem from the PN structure

Fig. 3 The PTPN corresponding to the sequences considered in Ex. 1



and timing specification. On the one hand, the value of the optimal solution of the LP problem is the upper/lower bound value of the performance metric of interest (e.g., upper bound mean throughput of a transition, lower bound mean time between two consecutive firings of a transition). However, the optimal solution of the LP problem provides useful information for the identification of critical paths in the model, e.g., the slowest subnet.

However, from a structural perspective, the semantics of PTPNs differ from those of TPNs and SPNs, making it impossible to directly apply existing bounding techniques to PTPN models. To address this, we adopt a transformation-based approach that enables the application of performance bounding techniques to PTPNs. This method involves the following steps:

Step 1 (pre-processing). Transform the PTPN model into an equivalent SPN model, associating continuous-time distributions with transitions.

Step 2 (solving). Apply SPN bounding techniques to the transformed model to compute performance bounds, such as throughput and inter-firing times.

Step 3 (post-processing). Map the computed results, including bound values and the optimal solution of the LP problem, back to the original PTPN model.

Step 1 - Pre-processing . Given a marked PTPN model, where each transition is defined by a continuous probability distribution associated with its firing delay, the pre-processing step involves transforming the PTPN into a semantically equivalent SPN model. This transformation replaces the forward incidence mapping of each PTPN transition ($Post$) with a corresponding SPN subnet, as illustrated in Fig. 4 for a transition t_0.

The Alg. 3 specifies the transformation: a marked PTPN model is provided as input, and a marked SPN model $\langle P', T', Pre', Post', D', W, M_0\rangle$ is produced, where:

- P' is the set of places;
- $T' = T_{timed} \cup T_{imm}$ is the set of (timed and immediate) transitions;
- $Pre' : T' \to \mathbb{N}^{P'}$ is the backward incidence mapping;
- $Post' : T' \to \mathbb{N}^{P'}$ is the forward incidence mapping;
- $D : T_{timed} \to \mathcal{D}(\mathbb{R}_+)$ is a function that assigns a continuous probability distribution to each timed transition;
- $W : T_{imm} \to \mathbb{R}^+$ is a function that assigns a weight of each immediate transition;

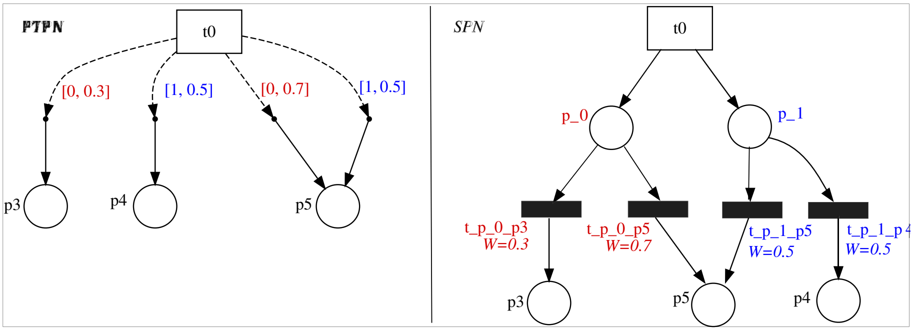


Fig. 4 Transformation pattern

- $M_0 : P' \to \mathbb{N}$ is a function that assigns an initial marking to each place.

Algorithm 3 PTPN to SPN transformation.

```
Require: (marked) PTPN: $\langle P, T, Pre, Post, D, \rho \rangle$
Ensure: (marked) SPN: $\langle P', T', Pre', Post', D', W, M_0 \rangle$
 1: initialize(SPN, PTPN)      ▷ $P' = P, T' = T, Pre' = Pre, Post' = \emptyset, D' = D, M_0 = \rho$
 2: for $t \in T$ do
 3:     $D_t$ = get_place_distributions($Post(t)$)      ▷ $D_t = \cup_{i=1}^{nt} \{d \mid d \equiv \cup_{j=1}^{ne} \{(p, pb_p) \mid p \in P\}\}$
 4:     $t'$ = get_transition_from_SPN($t$)
 5:     $t'^{\bullet} = \emptyset$
 6:     for $d \in D_t$ do
 7:         $P' = P' \cup \{p_d\}$
 8:         set_initial_marking($p_d$, 0)      ▷ $M_0(p_d) = 0$
 9:         $t'^{\bullet} = t'^{\bullet} \cup \{p_d\}$
10:         for $(p, pb_p) \in d$ do      ▷ $p$ is a place and $pb_p$ its probability in the distribution $d$
11:             $T' = T' \cup \{t'_{t,d}\}$      ▷ The new transition $t'_{t,d}$ is immediate.
12:             set_weight($t'_{t,d}$, $pb_p$)      ▷ The firing weight of $t'_{t,d}$ is equal to $pb_p$
13:             $p_d^{\bullet} = p_d^{\bullet} \cup \{t'_{t,d}\}$
14:             $p'$ = get_place_from_SPN($p$)
15:             $^{\bullet}p' = {}^{\bullet}p' \cup \{t'_{t,d}\}$
16:         end for
17:     end for
18: end for
19: return marked SPN
```

The algorithm starts by initializing the SPN model: all the net elements are copied from the PTPN model but the *Post* (line 1). Each PTPN transition $t \in T$ is then processed (lines 2-18). Concretely, we retrieve the set of place distributions associated to the transition, D_t (line 3), that includes a set of elements, where each element is a set of pairs (p, pb_p): p is a place and pb_p is the probability of producing a token in p when firing t. We use each element $d \in D_t$ to add a new SPN subnet (lines 6-17). In particular, a new place p_d is added to the set of places of the SPN (line 7), that is an output place of the SPN transition t' —copy of t (line 9), and as many new immediate transitions are added to the set of transitions in the SPN as the number of pairs (p, pb_p) in the place distribution set d (lines 10-16). Each immediate transition $t'_{t,d}$ is characterized by a firing weight equal to the probability pb_p (line 12), the input place p_d (line 13) and the output place p' —copy of p (line 15).

Step 2 - solving The SPN model, obtained by applying the PTPN-to-SPN transformation Alg. 3, is used as input for this step. We resort to bounding techniques from literature (Campos and Silva 1992) to compute bounds of performance metrics, concretely:

- Upper bound of the mean throughput of a transition.
- Lower bound of the mean inter-firing time of a transition.

Bounding techniques are based on the derivation of a Linear Programming (LP) problem from the SPN structure and timing specification.

For self-containment, we report the LP problems stated for the computation of the two types of bound, considering a SPN marked model $\mathcal{SPN} = \langle P, T, Pre, Post, D, W, M_0 \rangle$ as defined in the previous step. In particular, we introduce the following notation used for the statement of the two LP problems:

- Pre, the pre-incidence matrix of size $|P| \times |T|$, corresponding to the backward incidence mapping function *Pre*;
- Post, the post-incidence matrix of size $|P| \times |T|$, corresponding to the forward incidence mapping function *Post*;
- $\mathbf{\Delta}$, the mean firing delay vector of size $|T_{timed}|$, where an entry $\mathbf{\Delta}[t]$ is the mean value of the continuous probability distribution $\mathcal{D}(t)$.
- w, the weight marking vector of size $|T_{imm}|$, corresponding to the weight function *W*;
- $\mathbf{M_0}$, the initial marking vector of size|*P*|, corresponding to the initial marking function M_0;

LP problem for upper bound mean throughput The upper bound mean throughput of a transition $t \in T$ is the optimal solution value of the LP problem:

$$LP_X : max \& x[t] \tag{1}$$

$$s.t. \& M[p] = \mathbf{M_0}[p] + \sum_{t \in T} \Big(\text{Post}[p,t] - \text{Pre}[p,t] \Big) \sigma[t], \forall p \in P \tag{2}$$

$$\sum_{t \in {}^\bullet p} \text{Post}[p,t] x[t] = \sum_{t \in p^\bullet} \text{Pre}[p,t] x[t], \forall p \in P \tag{3}$$

$$M[p] \geq \text{Pre}[p,t] \mathbf{\Delta}[t] x[t], \forall t \in T_{timed}, \forall p \in {}^\bullet t \tag{4}$$

$$\text{w}[t'] x[t] = \text{w}[t] x[t'], \forall t, t' \in T_{imm} \wedge \ ECS(t,t') = true \tag{5}$$

$$x[t], \sigma[t] \geq 0, \forall t \in T, M[p] \geq 0, \forall p \in P \tag{6}$$

where, the objective function Eq. 1 is the mean throughput variable associated to transition *t*; constraints Eq. 2 are the marking reachability equations; constraints Eq. 3 are the token flow equations; constraints Eq. 4 are the utilization law inequalities; constraints Eq. 5 are the routing equations defined by the firing weights of the immediate transitions belonging to the same extended conflict set (ECS). Finally, all the problem variables Eq. 6 are non negatives.

LP problem for lower bound mean inter-firing time An optimal solution of the LP problem LP_X (1-6) enables to compute the transition visit ratios, provided that $t \in T$ is live. Indeed, in such a case the upper bound mean throughput $x^*[t] \neq 0$ and the visit ratio vector is defined as $\text{v}[t'] = \frac{x^*[t']}{x^*[t]}$.

The lower bound mean inter-firing time of a transition $t \in T$ is the optimal solution value of the LP problem:

$$LP_{CT} : max \& \sum_{p \in P} \sum_{t \in T} \big(\text{Pre}[p,t] \mathbf{\Delta}[t] \text{v}[t] \big) \ y[p] \tag{7}$$

$$s.t.\& \sum_{p \in P} \Big(\mathrm{Post}[p,t] - \mathrm{Pre}[p,t] \Big) y[p] = 0, \forall t \in T \tag{8}$$

$$\sum_{p \in P} \mathbf{M_0}[p] y[p] = 1 \tag{9}$$

$$y[p] \geq 0, \forall p \in P \tag{10}$$

where, the objective function Eq. 7 is the cycle time of a token in a circuit of the SPN model, where the circuit is defined by the support of a P-semiflow *y*. Constraints Eq. 8 enable to identify the P-semiflows of the SPN model and constraints Eq. 9 select the normalized P-semiflows. A P-semiflow *y* is characterized by non-negative values (constraints Eq. 10).

Observe that optimal solutions y^* of LP_{CT} enable identifying the slowest subnets of the SPN: all the transitions belonging to such subnets (including *t*) are characterized by the same mean inter-firing time (i.e., value of the optimal solution).

Step 3 - Post-processing This step synthesizes the results from the optimal solutions of the LP problems LP_X and LP_{CT} and maps them back to the original PTPN model.

Let $\mathcal{PTPN} = \langle P, T, Pre, Post, D, \rho \rangle$ and $\mathcal{S}PN = \langle P', T', Pre', Post', D', W, M_0 \rangle$ be a PTPN model and the SPN obtained from the Alg. 3, respectively. We denote as $\phi : P \rightarrow P'$ and $\psi : T \rightarrow T'$, the injective mapping functions of the transformation pattern.

***LP_X* problem solution** Given a transition $t \in T$, an LP_X problem is formulated considering its SPN counterpart, that is $t' = \psi(t) \in T'$. An optimal solution of the LP problem is a triplet of vectors (x^*, σ^*, M^*), where $x^*[t']$ is the optimal value. Such value represents the upper-bound mean throughput of t' and, thus, of $t = \psi^{-1}(t')$.

***LP_{CT}* problem solution** Given a transition $t \in T$, an LP_{CT} problem is formulated considering its SPN counterpart, that is $t' = \psi(t) \in T'$. In particular, the visit ratio vector v is computed with t' as reference transition, i.e., $\mathrm{v}[tr] = \frac{x^*[tr]}{x^*[t']}, \forall tr \in T'$. An optimal solution of the LP problem is a vector y^* and the slowest subnet of the SPN, identified by $y*$, is thus characterized by the following places and transitions:

$$P'_{slowest} = ||y^*|| = \{p' \in P' \mid y^*[p'] > 0\} \tag{11}$$

$$T'_{slowest} = ^{\bullet}P'_{slowest} \cap P'^{\bullet}_{slowest} \tag{12}$$

The subnet can be traced back to a subnet in the original PTPN:

$$P_{slowest} = \{p \in P | \phi(p) \in P'_{subnet}\} \tag{13}$$

$$T_{slowest} = \{t \in T | \psi(t) \in T'_{subnet}\} \tag{14}$$

where all the transitions $t \in T_{slowest}$ have the same lower bound interfiring time (i.e, the optimal solution value of LP_{CT}).

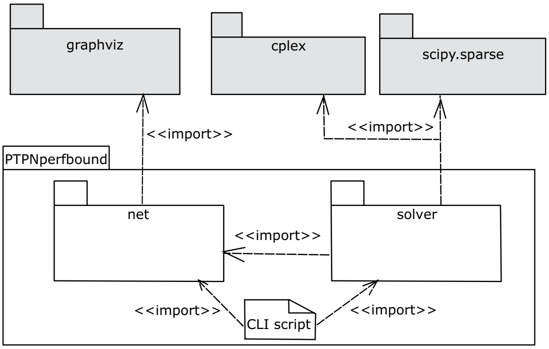


Fig. 5 Architecture

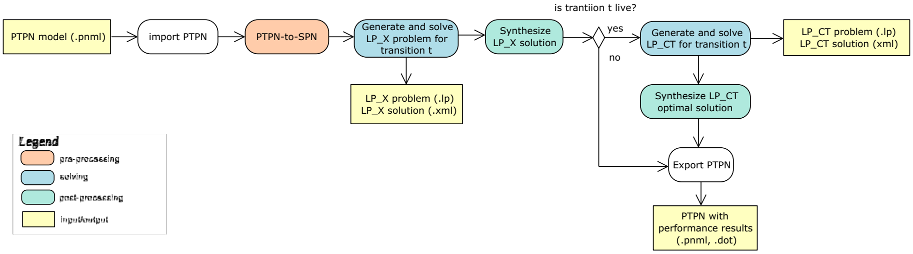


Fig. 6 Workflow

A PTPN performance bound tool A tool for the calculation of the PTPN performance limits has bee integrated in out implementation, following the reduction-based approach described previously. The tool is structured in two main packages (Fig. 5): net, including the modules responsible of importing/exporting the PTPN models; and solver, including the modules responsible of transforming the PTPN model to an equivalent SPN model, generating the LP problems (LP_X and LP_{CT}), solving the LP problem and synthesizing the results. Besides, a command-line interface (CLI) script is the tool entry-point from the user point of view. The tool relies on external Python packages: in particular, the APIs of the cplex package have been used for the generation and solution of the LP problems.

Figure 6 shows the main workflow steps of the tool. In particular, after importing the PTPN model and the pre-processing step, where the PTPN model is transformed to a SPN model, there are two phases. In the first phase, the problem LP_X for the transition *t* of interest is generated and solved, and accordingly, the PTPN model is updated with the solution (i.e., the upper bound mean throughput of *t*). The second phase is carried out in case the transition is live (i.e., $x^*[t] \neq 0$), thus the LP_{CT} problem for *t* is feasible and it is generated and solved, and then the PTPN model is updated with the solution (i.e., the lower bound mean interfiring time associated of *t* and the slowest subnet). The tool enables to export the PTPN with the solutions in pnml and dot formats.

PTPN models interchange format The tool implements the import/export functionalities through XML deserialization/serialization. In particular, an XML schema is defined for PTPN that is compliant with the standard PNML for Place-Transition nets (PNML n.d.).

Figure 7 shows the metamodel of the PTPN interchange format, where the white classes are from the PNML metamodel and the colored classes represent the tool-specific extensions. The yellow ones are core concepts of the PTPN formalism, i.e., discrete probability distributions over output places of a transition and continuous probability distributions associated to transition delays. The blue classes are concepts related to the results obtained from the performance bound solver, i.e., the performance bounds and the slowest subnet.

6 Case study: Anterior Cruciate Ligament (ACL) rupture

The Anterior Cruciate Ligament (ACL) is an intraarticular ligament of the knee, acting as a stabilizer by controlling the forward movement of the tibia relative to the femur during knee flexion and extension. It consists of two bundles: the anteromedial bundle, which controls forward translation, and the posterolateral bundle, which limits internal rotation and pivot shifts. ACL injuries occur at an estimated rate of 0.4 to 0.8 per 1,000 people, annually, among those aged 10 to 64 (Gianotti et al. 2009), a number that is increasing with increased sports participation. This rise has led to more surgical interventions, facilitated by advancements in techniques and outcomes. Inappropriate repair of the ACL can severely affect quality of life, particularly in athletes, and may lead to knee joint degeneration over time if left untreated.

ACL ruptures are common in athletes and typically result from sudden directional changes, abrupt stops, or hyperextension during physical activity. Standard treatment involves surgical reconstruction followed by a structured rehabilitation program. Previously, no formal clinical pathway was established for ACL injuries at the Hospital Clínico

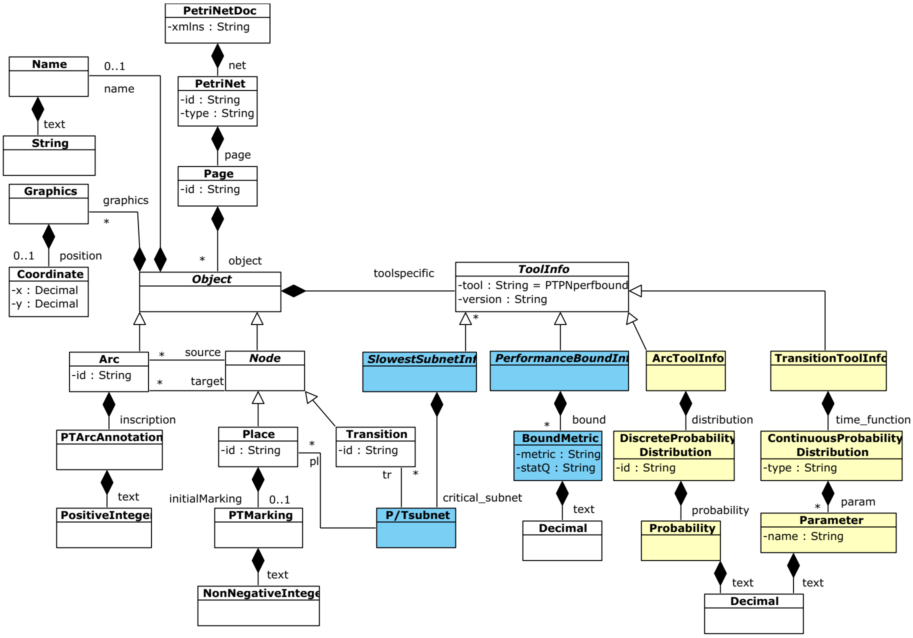


Fig. 7 PTPN interchange format meta-model

Universitario (HCU) in Zaragoza. However, with the collaboration of medical professionals and the methodology described in this work, a new clinical pathway was developed to standardize treatment and rehabilitation.

A dataset of 50 patients, who underwent ACL rupture surgery in the hospital, was used as input for this study. This dataset includes all the activities performed during the clinical treatment process in the hospital. The data were analyzed using pattern recognition algorithms to derive a PTPN representing the clinical pathway. The clinical pathway was iteratively refined with continuous feedback from medical doctors to address data gaps and resolve inaccuracies. The modeling process used the pattern discovery algorithm (see Section 3) and the PTPN methodology described in Section 4, providing an accurate representation of the treatment pathway while accommodating variations in patient outcomes.

Initial data collection and first model The initial data for this case study were collected manually from hospital records by the medical team at the Hospital Clínico Universitario (HCU) in Zaragoza. Since no automated system was available, the data were transcribed into a spreadsheet, anonymized to protect patient privacy, and then preprocessed for analysis. This data set included anonymized demographic data for the patient, surgical details, postoperative care, and rehabilitation activities. Once the data were processed, an initial PTPN model was generated using a pattern discovery algorithm to represent the typical sequence of activities in the ACL treatment pathway.

Iteration 1: Model validation and data corrections During the first iteration of model validation with the medical team, several critical errors were identified. These errors were primarily due to missing or incorrectly recorded data in the original spreadsheet. The following issues were addressed:

- *Incorrect start and end points of the model*. The ’Discharge’ activity was incorrectly placed before the completion of intermediate steps like ’Consent for surgery’. This mistake was traced to incomplete data entries in the spreadsheet, where the start and end points of some patient pathways were either missing or incorrectly recorded. After revising the data, the sequence was corrected to ensure ’Discharge’ occurred only after all necessary activities were completed.
- *Misordering of activities*. The activities ’Drain removal on the first day’ and ’Ambulation’ were found to be out of order. According to clinical practice, drain removal should always precede ambulation. However, the initial spreadsheet showed these activities in reverse order for several patients. After discussions with the medical team, the order of activities in the dataset was corrected to reflect the correct clinical sequence, and the model was updated accordingly.
- *Naming confusion*. The ’Antibiotic treatment’ activity was inaccurately labeled in the dataset as ’AB Treatment’, leading to confusion about its role in the pathway. This was clarified by renaming the activity to ’In-hospital Prolongation of Antibiotic Treatment’, based on clinical guidelines. The correction in the spreadsheet allowed for clearer identification of this activity in the PTPN.
- *Post-surgery activity disorder*. The activity ’HBPM post-surgery’ (administration of heparin) was incorrectly placed after antibiotic treatment in the dataset, when it should occur immediately after surgery. This mistake was due to missing timestamps in the data

entries, which caused incorrect sequencing. Once corrected in the spreadsheet, the order of activities in the model was fixed to accurately represent the post-surgery process.

Iteration 2: Refine the model and additional corrections to the data After addressing the initial errors, further refinements were introduced based on additional rounds of feedback from the medical team. These refinements correct incomplete or incorrectly interpreted information in the dataset:

- *Missing activity 'Wound dressing check'*. Some patients were noted to be missing the 'Dressing wound check' activity in the initial model. This was due to inconsistent data entry, where some patient records lacked the required inspection. To correct this, a new activity, 'Surgical wound inspection', was added to ensure that all patients undergo a wound check post-surgery, regardless of the outcome of the dressing inspection. This correction was reflected in the updated spreadsheet.
- *Ambiguity in antibiotic continuation*. In some cases, patients who showed signs of wound complications did not receive antibiotic prolongation, despite the need for treatment. Upon reviewing the data with the medical team, it was determined that the antibiotic continuation was often left unrecorded in the spreadsheet. The dataset was updated to ensure that the 'Prolongation of antibiotics' activity followed the correct decision-making process based on wound inspection results.

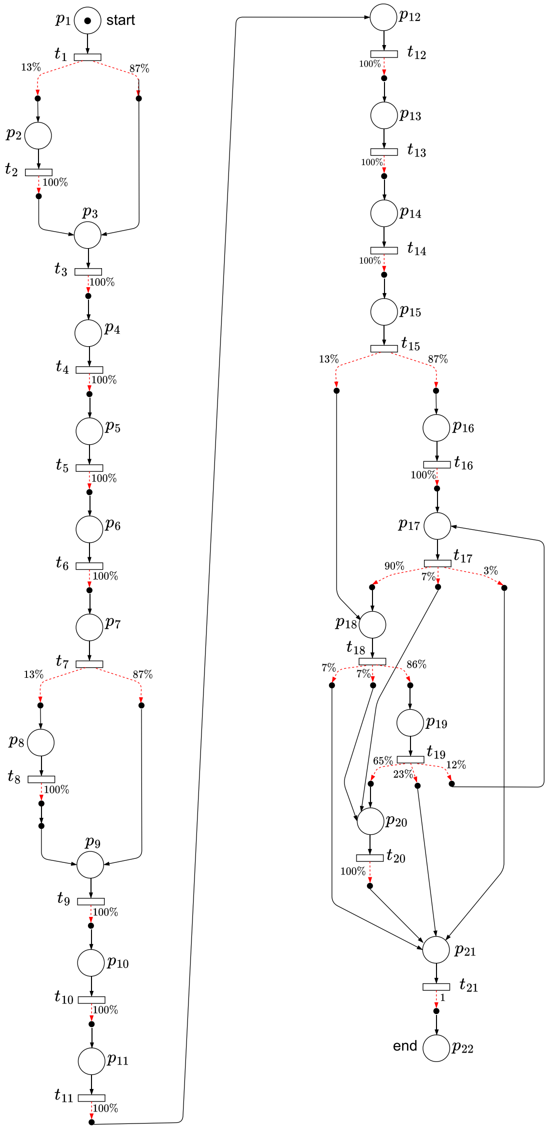
Iteration 3: Final model and validation In the third iteration, the model underwent a final round of validation, which revealed an important error concerning the ambulation process:

- *Skipping ambulation in some patients*. The model indicated that a small percentage of patients (13%) skipped the 'Ambulation' process after surgery. This error was traced back to missing entries in the spreadsheet, where ambulatory activity was not recorded for certain patients. After reviewing medical records, it was confirmed that all patients underwent ambulation as part of their rehabilitation process. The spreadsheet was updated to include missing ambulation entries, ensuring that all patients followed the correct clinical pathway.

Final results The final version of the PTPN accurately models the clinical pathway for ACL treatment, incorporating corrections made to the original patient dataset. The model is presented in Table 2 (note that a part of the database has been used in Sec. 3 and the partial model in Fig. 3, obtained by the discovery algorithm, is different from this one, since only three patients and partial logs have been used).

Although the current PTPN incorporates probabilistic branching, it does not yet include activity time delays due to the absence of recorded duration data in the dataset. Nonetheless, this limitation can be addressed using the methodology presented in this paper, which supports the integration of continuous-time distributions. This enhancement would allow the model to capture both variability in patient outcomes and the duration of each clinical activity, thereby providing valuable insights into resource allocation, patient care, and overall efficiency. The iterative approach, conducted in close collaboration with medical

Table 2 The PTPN corresponding to the pre-admission and admission days of the ACL pathway



Transition names:

- t_1: Pre-admission MRI
- t_2: Request for allograft before admission
- t_3: Surgical consents
- t_4: Pre-anesthetic evaluation in consultations
- t_5: Verification of anesthesia evaluation-hospital request
- t_6: Medication regimen and antibiotic induction
- t_7: Verification of surgical informed consent
- t_8: Verification of informed consent for allograft
- t_9: Verification of fasting for surgery
- t_{10}: Antibiotics during surgery
- t_{11}: Surgical protocol
- t_{12}: Low molecular weight heparin post-surgery
- t_{13}: Antibiotics post-surgery
- t_{14}: Removal of drain on the first day
- t_{15}: Checking the surgical wound
- t_{16}: Dressing Staining
- t_{17}: In-hospital extension of antibiotics
- t_{18}: Ambulation
- t_{19}: Rehabilitation
- t_{20}: Analgesic rescue
- t_{21}: Discharge

professionals and through continuous data refinement, ensures the model accurately reflects clinical realities and is suitable for practical application.

Obtained clinical pathway description The clinical pathway obtained begins in an outpatient setting when the patient is evaluated, necessary diagnostic tests are requested (t_1: 'Pre-admission_MRI'), and informed consent is obtained. Verbal information is provided during the specialist consultation, but written confirmation is required to ensure the patient fully understands the procedure (t_3: ’Surgical_consents’). At this stage, surgical materials are requested, including allografts from the Aragón Blood and Tissue Bank (t_2: ’Request_for_allograft_before_admission’). In addition, the patient undergoes a pre-anesthetic evaluation by the anesthesia team (t_4: ’Pre-anesthetic_evaluation_in_consultations’).

Upon hospital admission, the medical team verifies the completion of the pre-anesthetic evaluation (t_5: ’Verification_of_anesthesia_evaluation-hospital_request’), ensures the availability of surgical materials, and reviews the necessary documentation, such as surgical consents and informed consent for allografts (t_7: ’Verification_of_surgical_informed_consent’, t_8: ’Verification_of_informed_consent_for_allograft’). Medication is prescribed for the surgery day, including antibiotics for prophylaxis and fasting instructions (t_6: ’Medication_regimen_and_antibiotic induction’, t_9: ’Verification_of_fasting_for_surgery’).

On the day of the surgery, antibiotics are administered during the pre-surgical induction phase (t_{10}: ’Antibiotics_during_surgery’). After the surgical procedure, the surgical protocol is documented in the patient’s clinical history (t_{11}: ’Surgical_protocol’), and post-surgical medications, including antibiotics and low molecular weight heparins, are prescribed (t_{13}: ’Antibiotics_post-surgery’, t_{12}: ’Low_molecular_weight_hepari_post-surgery’).

During the first postoperative day, intra-articular drains are removed (t_{14}: ’Removal_of_drain_on_the_first_day’), and the surgical wound is examined (t_{15}: ’Checking_the_surgical_wound’). If dressing staining is observed (t_{16}: ’Dressing_Staining’), the antibiotic prophylaxis is extended according to protocol (t_{17}: ’In-hospital_extension_of_antibiotics’). Most patients are allowed to walk (t_{18}: ’Ambulation’), although specific cases may require restricted weight-bearing, such as when additional surgical procedures, such as meniscal sutures, are performed.

A key quality objective of the pathway is to ensure that patients are evaluated by the rehabilitation service (t_{19}: ’Rehabilitation’). This enables them to begin outpatient physiotherapy or follow prescribed guidelines if rehabilitation is not required. Analgesic treatment is provided during the immediate postoperative period, with rescue medication available upon patient request. Approximately 60% of patients utilize this option (t_{20}: ’Analgesic_rescue’).

The ultimate goal of the clinical pathway is to discharge the patient efficiently (t_{21}: ’Discharge’), optimizing material and personnel resources while minimizing costs, maintaining high-quality care standards, and ensuring patient satisfaction.

The Wilson confidence interval can be employed in sensitivity analysis to evaluate the robustness of transition probabilities in the PTPN model. By varying these probabilities within the Wilson intervals and observing the resultant changes in key performance metrics, such as throughput, analysts can assess how uncertainty in transition probabilities impacts the clinical pathway outcomes. This provides valuable insights into the stability and reliability of the model under different scenarios of uncertainty.

7 Conclusions

This paper presents a methodology for modeling and analyzing clinical pathways using Probabilistic Timed Petri Nets (PTPN). By combining formal methods with real-world data, the proposed approach provides a structured and probabilistic framework for optimizing healthcare workflows. The iterative refinement process, conducted in collaboration with medical professionals, ensures that the models developed accurately represent clinical practices while addressing data inconsistencies. The integration of continuous-time distributions further enhances the model's ability to simulate real-world variability, enabling performance evaluations and resource optimization.

This work will help optimize the use of limited healthcare resources while standardizing outcomes regardless of the professionals involved in treatment. Employing clinical pathways based on the best available scientific evidence offers clear economic, healthcare, and legal advantages. Specifically, this methodology reduces unnecessary variations in care, minimizes resource misuse, and ensures compliance with established medical standards, ultimately improving both patient outcomes and the efficiency of healthcare systems.

Future work will focus on incorporating real-time data integration to dynamically update clinical pathways and expanding the methodology to other pathologies and resource-intensive fields. In addition, a web-based application for hospital management is envisioned to further streamline resource allocation and clinical workflow monitoring. This study underscores the potential of PTPNs to transform healthcare delivery by bridging the gap between formal models and practical applications.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10626-025-00419-4.

Acknowledgements Dr. Juan Falcón Goicoechea, resident doctor in the Department Orthopedic Surgery and Traumatology at the Clinical Hospital of Zaragoza, for his invaluable assistance with data collection. The authors also thank Marcos Vázquez for his dedicated work on the case study as part of his final project in collaboration with the medical team.

Author Contributions M.L.M. (Manon Le Moigne) implemented the pattern discovery algorithm, extended it to incorporate time distributions, and prepared the initial draft of the manuscript. C.M. (Cristian Mahulea) supervised M.L.M., facilitated interactions with medical professionals, revised the manuscript, integrated the new case study, executed the corresponding simulations, and edited the figures. G.F. (Gregory Faraut) supervised M.L.M.'s implementation of the pattern discovery algorithms and contributed to manuscript revisions. S.B. (Simona Bernardi) implemented and integrated the performance evaluation framework for computing bounds and revised the manuscript. J.A. (Jorge Albareda) and L.C. (Lidia Castán) were responsible for the medical aspects and provided key contributions to the case study.
All authors reviewed and approved the final version of the manuscript.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data Availability The clinical data supporting the findings of this study are available and can be provided upon reasonable request. The data have been anonymized to ensure patient confidentiality and privacy. Interested researchers can contact the corresponding author at cmahulea@unizar.es to request access to the data.

Declarations

Competing interests The authors declare no competing interests.

References

Aalst WMP, Carmona J (eds) (2022) Process mining handbook 448. Series: Lecture Notes in Business Information Processing, Springer

Aalst WMP (2016) Process Mining: Data Science in Action, 2nd edn. Springer. https://doi.org/10.1007/978-3-662-49851-4

Aalst WMP (2010) Process discovery: Capturing the invisible. IEEE Comput Intell Mag 5:28–41

Aalst W, Dongen B, Günther C, Rozinat A, Verbeek E, Weijters A (2009) ProM: The Process Mining Toolkit. Allergy

Bernardi S, Campos J (2013) A Min-Max Problem for the Computation of the Cycle Time Lower Bound in Interval-Based Time Petri Nets. IEEE Trans Syst Man Cybern Syst 43(5):1167–1181. https://doi.org/10.1109/TSMCA.2012.2226442

Bernardi S, Mahulea C, Albareda J (2019) Toward a decision support system for the clinical pathways assessment. Discrete Event Dynamic Systems: Theory and Applications 29(1):91–125

Berti A, van Zelst S, Schuster D (2023) Pm4py: A process mining library for python. Softw Imp 17:100556. https://doi.org/10.1016/j.simpa.2023.100556

Campos J, Silva M (1992) Structural techniques and performance bounds of stochastic Petri net models. In: Rozenberg G. (ed.) Advances in Petri Nets 1992, The DEMON Project. Lecture Notes in Computer Science, vol 609, pp 352–391. Springer,. https://doi.org/10.1007/3-540-55610-9_178

Emzivat Y, Delahaye B, Lime D, Roux OH (2016) Probabilistic time Petri nets. 37th international conference on applications and theory of Petri nets and concurrency, pp 261–280

Fouquet K, Faraut G, Lesage J-J (2020) Life habits modeling with stochastic timed automata in ambient assisted living. In: 2020 IEEE international conference on systems, man, and cybernetics

Gianotti SM, Marshall SW, Hume PA, Bunt L (2009) Incidence of anterior cruciate ligament injury and other knee ligament injuries: A national population-based study. J Sci Med Sport 12(6):622–627

Hindle D, Yazbeck A, Mills PD (2006) Patient pathways: A tool to enhance quality and efficiency in health care. Aust Health Rev 30(3):372–380

Le Moigne T, Mahulea C, Faraut G (2024) Optimizing Clinical Pathways: A Probabilistic Time Petri Net Approach. IFAC-PapersOnLine 58(1):96–101. WODES 2024: 17th IFAC Workshop on discrete Event Systems

Mahulea C, Mahulea L, Garcia-Soriano JM, Colom JM (2018) Modular Petri net modeling of healthcare systems. Flex Serv Manuf J 30:329–357

Mould G, Bowers J, Ghattas M (2010) The evolution of the pathway model. Int J Care Pathw 14(4):142–150

PNML interchange format. https://www.pnml.org/

Rifki O, Peng Z, Perrier L, Xie X (2024) Process mining with event attributes and transition features for care pathway modelling. Int J Prod Res, pp 1–25

Saives J, Pianon C, Faraut G (2015) Activity discovery and detection of behavioral deviations of an inhabitant from binary sensors. IEEE Trans Autom Sci Eng 12(4):1211–1224

Saives J, Faraut G (2014) Automated generation of models of activities of daily living. WODES 2014 - 12th International Workshop on Discrete Event Systems, pp 13–20

Vanhaecht K, Panella M, Zelm R, Sermeus W (2010) An overview on the history and concept of care pathways as complex interventions. Intl J Care Path 14(4):117–123

Weijters AJMM, Aalst WMP, Medeiros AKA (2006) Process mining with the heuristics miner-algorithm. Technical report, Technische Universiteit Eindhoven

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Manon Le Moigne is a Ph.D. student at Centrale Nantes, Nantes Université, France. She is graduated from ENS Paris-Saclay, Université Paris-Saclay, France in 2025. The work for this paper was done during her pre-doctoral research year at Zaragoza University, Spain. Her research interests include identification of discrete-event systems and Petri nets.

Cristian Mahulea received his B.S. and M.Sc. degrees in control engineering from the Technical University of Iasi, Romania, in 2001 and 2002, respectively, and his Ph.D. in systems engineering from the University of Zaragoza, Spain, in 2007. Currently, he is a Full Professor at the University of Zaragoza, where he chaired the Department of Computer Science and Systems Engineering from 2020 to 2024. He has also served as a visiting professor at the University of Cagliari, Italy. His research interests include discrete event systems, hybrid systems, mobile robotics, and healthcare systems. He has been a Visiting Researcher at the University of Sheffield (UK), Boston University (USA), University of Cagliari (Italy), and ENS Paris-Saclay (France). Cristian has served as an Associate Editor for IEEE Transactions on Automation Science and Engineering (TASE), IEEE Control Systems Letters (L-CSS) and IEEE Robotics and Automation Letters (RA-L). He is currently an Associate Editor for IEEE Transactions on Automatic Control (TAC), International Journal of Robotics Research (IJRR) and Discrete Event Dynamic Systems: Theory and Applications (JDES). Additionally, he was the General Chair of ETFA 2019.

Gregory Faraut received the B.S. degree in electrical engineering and the M.S. degree in computer science from the University of Nice Sophia Antipolis, Nice, France, in 2004 and 2006, respectively, and the Ph.D. degree in automatic control from the Ampere Lab, INSA Lyon, Villeurbanne, France, in 2010. Since 2011, he has been an Associate Professor of Automatic Control with LURPA, ENS Paris-Saclay, University of Paris-Saclay, and Full Professor since 2022. His research interests concern the field of Discrete Event Systems with applications to human-cyber-physical systems, behavioral identification, resilient control, cognitive systems, and, more recently, Digital Twins.

Simona Bernardi is an Associate Professor in the Department of Computer Science and Systems Engineering at the University of Zaragoza, Spain. She received a MS degree in mathematics and a PhD degree in computer science, in 1997 and 2003, respectively, both from the University of Torino, Italy. Her research interests are in the area of software engineering, in particular model driven engineering, verification and validation of performance, dependability and survivability software requirements, and formal methods for the modelling and analysis of software systems. She is also interested in the application of anomaly detection techniques in energy frauds. She has served as a referee for international journals and as a program committee member for several international conferences and workshops.

Dr. Jorge Albareda is the Emeritus Head of Department of Orthopedic Surgery of de University Hospital "Lozano Blesa" of Zaragoza (Spain) and Associated Professor of Orthopedic Surgery, University of Zaragoza (Spain), Head of Studies and President of the Teaching Commision of the University Hospital "Lozano Blesa" of Zaragoza (Spain).

Dr. Lidia Castán completed her orthopedic surgery and traumatology training at Hospital Clínico Universitario Lozano Blesa, earning her specialty certification in May 2010. She then served as a specialist physician at Hospital Nuestra Señora de Gracia and later in the hip and knee arthroplasty unit at Hospital Royo Villanova until 2017, when she returned to the Lozano Blesa Clinical University Hospital. An expert in hip and knee surgery, she currently practices in the Knee Unit at Lozano Blesa and is an active member of SECOT and AEA. In 2013, she augmented her training with an AAOS continuing education course in Chicago, and from 2017 to 2019 she completed Spain's recertification program in orthopedic surgery and traumatology. Her honors include a 2016 research fellowship from the Zaragoza Medical College and a Cum Laude doctoral thesis on sarcopenia, nutrition, and osteoporosis in elderly femur-fracture patients. She earned a 2014 master's in medical evaluation and expert testimony from the University of Barcelona and has collaborated on undergraduate and postgraduate teaching since 2006, including courses for nursing staff, medical students, and residents. Author and coordinator of multiple book chapters and editorials, she has presented over 50 posters and numerous papers at regional, national, and international congresses.

Authors and Affiliations

Manon Le Moigne[1] · Cristian Mahulea[2] · Gregory Faraut[1] · Simona Bernardi[2] · Jorge Albareda[3] · Lidia Castán[3]

✉ Cristian Mahulea
cmahulea@unizar.es

Manon Le Moigne
manon.le_moigne@ens-paris-saclay.fr

Gregory Faraut
gregory.faraut@ens-paris-saclay.fr

Simona Bernardi
simonab@unizar.es

Jorge Albareda
jcalbareda@salud.aragon.es

Lidia Castán
lcastan@salud.aragon.es

[1] ENS Paris-Saclay, University of Paris-Saclay, Gif-sur-Yvette, France

[2] Aragon Institute for Engineering Research (I3A), University of Zaragoza, Zaragoza, Spain

[3] Hospital Clínico Universitario (HCU) “Lozano Blesa” in Zaragoza, Zaragoza, Spain