

Summary of “Automatically reconciling the trade-off between prediction accuracy and earliness in prescriptive business process monitoring.” -

Andreas Metzger, Tristan Kley, Aristide Rothweiler, Klaus Pohl

Section 1: Introduction

- **Predictive process monitoring** attempts to answer “what will happen and when?”.

It uses data from the ongoing execution of a business process instance (a.k.a. case), as well as historical data, to forecast the future state of the case. It can continuously generate predictions as new data arrives from an ongoing case, but it provides limited support to process managers regarding the good and bad.

- **Prescriptive process monitoring** attempts to answer “when to intervene and how?”.

It helps process managers decide when to intervene and adapt to an ongoing case. With “alarms,” it assists process managers in taking action. However, if raised too early, it could impact accuracy due to the limited data collected from the ongoing case.

Section 2: Fundamentals

Predictive business process monitoring forecasts how a case or instance will progress, based on event logs. An event has at least a categorical attribute (event type) and a numerical attribute (timestamp). The resource that has carried out the activity is considered.

Informally, prediction accuracy aims to predict as many true deviations from expected process outcomes as possible, while minimizing false deviations. This is to avoid unnecessary adaptation and missed opportunities for adaptation.

Predictions occur at various points during a case's execution. Early predictions are beneficial as they provide more time to address anticipated deviations. Prediction points can be defined by identifying relevant activities, milestones, or using equidistant time points. Typically, predictions are made after each event of an ongoing case. We identify these prediction points by their prefix length, indicating the number of events preceding the prediction.

The cost-based model takes these parameters into account:

- Penalty costs: cost of undesired outcome
- Adaptation costs: cost of intervention
- Adaptation effectiveness: mitigation effectiveness
- Compensation costs: cost of compensation

Section 3: Problem Statement

Early Classification of Time Series (ECTS) deals with time series data, which represents the values of variables at different points in time (like temperatures or stock prices). **Business Process Management (BPM)** uses process monitoring data, which are sequences of process events. These events are characterized by timestamps, indicating when they occurred, such as the completion of a process activity.

Section 4: State-of-the-art approaches

First Positive Prediction under the *selected approach*: easy to implement, but it ignores the accuracy of data, so it may lead to false alarms.

Static prediction point: Using a fixed prediction point might not trigger any alarm if the length is shorter than the condition. Plus, it doesn't provide direct information about the accuracy of an individual prediction for a case.

Thresholding: A **reliability estimate** quantifies the probability that a specific prediction is accurate.

Empirical thresholding: A recent BPM approach uses a dedicated training process and dataset to determine a suitable threshold. The basic variant employs a cost model defining adaptation, compensation, and penalty costs to calculate the optimal threshold.

Online Reinforcement Learning (Online RL): This method learns during the actual execution of business processes, making decisions based on predictions and their reliability. It adjusts its decision-making process continuously based on the accuracy of past decisions, avoiding the need for fixed thresholds or calibration of reliability estimates, unlike Empirical Thresholding.

Section 5: Online RL with artificial curiosity

Action Selection and Policy: This involves directly optimizing a stochastic action selection policy represented by a neural network. It selects actions based on probabilities, allowing it to handle multi-dimensional continuous state spaces and generalize well over unseen states. Unlike value-based deep RL, policy-based deep RL can readily address concept drifts without needing explicit exploration-exploitation strategies, making it suitable for prescriptive process monitoring, especially in dynamic environments.

Policy Update: Using policy gradient methods, the neural network's weights are adjusted based on past actions, states, and rewards. This optimization aims to maximize an objective function, such as average rewards over time.

Reward Function: This determines the numeric reward received for taking action. It quantifies the learning objective. The key challenge is designing the reward function to ensure that maximizing cumulative rewards aligns with solving the underlying problem.

States and Actions: States refer to the output from the prediction model. Actions are binary, indicating whether to raise or not raise an alarm at a given prediction point.

Learning Episodes: For each executed case or when an alarm is raised, it gets a reward. Otherwise, there is no reward.

Section 6: Evaluation setup and execution

Random forest prediction models are trained in R using the 'randomForest' package. They require encoding sequential event sequences into fixed-length inputs. To maintain comparability with LSTM models and avoid tuning additional parameters, we use a two-step encoding process:

- **Trace Bucketing**: Event sequences are divided based on prefix length into buckets, each with a trained random forest model.
- **Sequence Encoding**: Event sequences within buckets are transformed into fixed-length vectors using index encoding, preserving all information and sequence length consistency.

We adapted individual **LSTM models**, traditionally used for predicting subsequent activities, to directly forecast the process outcome. Direct prediction methods have been shown to be more accurate than incremental ones.

Section 7 & 8: Selection and characterization of experimental data sets / Results

A. Metzger, T. Kley, A. Rothweiler et al.

Information Systems 118 (2023) 102254

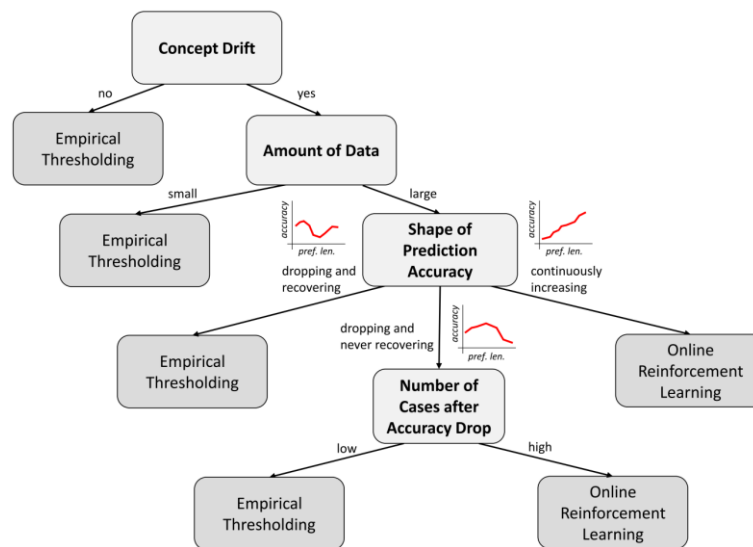


Fig. 11. Decision tree providing recommendations to decide between alternative approaches.