# Predictive Performance Monitoring of Material Handling Systems Using the Performance Spectrum

Vadim Denisov
Eindhoven University of Technology
The Netherlands
Email: v.denisov@tue.nl

Dirk Fahland
Eindhoven University of Technology
The Netherlands
Email: d.fahland@tue.nl

Wil M.P. van der Aalst
Department of Computer Science,
RWTH Aachen, Germany,
Email: wvdaalst@pads.rwth-aachen.de

***Abstract*—Predictive performance analysis is crucial for supporting operational processes. Prediction is challenging when cases are not isolated but influence each other by competing for resources (spaces, machines, operators). The so-called performance spectrum maps a variety of performance-related measures within and across cases over time. We propose a novel prediction approach that uses the performance spectrum for feature selection and extraction to pose machine learning problems used for performance prediction in non-isolated cases. Although the approach is general, we focus on material handling systems as a primary example. We report on a feasibility study conducted for the material handling systems of a major European airport. The results show that the use of the performance spectrum enables much better predictions than baseline approaches.**

## I. Introduction

*Predictive Process Monitoring* (PPM) is crucial for supporting the operation of *Material Handling System* (MHS), such as *Baggage Handling Systems* (BHS) of airports, where undesired or unexpected performance scenarios can lead to congestion, inefficient management of manual operations, baggage mishandling (e.g. being late for a flight) and as a result, to lower customer satisfaction and higher operational costs [1]. Predictive analysis of *Process Performance Indicators* (PPI), which can reveal such a problematic scenario in advance allowing to take mitigating actions. While classically the problem is addressed through manually built simulation models  [2], [3], also event data generated by material handling processes are available describing movement of materials or manual operations in the past. Current Machine Learning (ML) approaches for PPM in MHS [4] make limited use of the process dimension in the data. In recent years a variety of PPM approaches have appeared for Business Processes (BP) [5]. However, these techniques mostly assume isolated cases and stationary processes, these assumptions are violated for MHS [6]. Recently proposed PPM approaches for business processes [5] focus on predicting a single case assuming isolated cases and stationary processes. On one hand, these assumptions are violated for MHS and most business processes [6]. The performance of each case is dependent on the performance of the cases "around", i.e., the recent state of the system itself and the recent performance of handling groups of cases, rather than on an individual case performance or properties. In such cases, for the instance inter-case similarity-based features are required for predicting remaining time until case completion [7]. On the other hand, relevant PPIs in practice [5], including MHS, are not measures for an individual case but *aggregate* measures such as the amount of work (cases) expected or the occurrence of high waiting times (for all cases) at a particular step in a specific time-window from now. In this paper, we consider the problem of predicting aggregate PPM in (non-stationary) processes competing for shared resources, e.g., baggage handling in an MHS under changing load. In the following we refer to this as the *Inter-Case Performance Prediction Problem* (IC3P). The *Performance Spectrum* (PS) [6] formats performance data from process event logs in a way that maps the performance of each case over each process step over time. The PS reveals the performance of all cases in a step in relation to preceding and succeeding steps, non-stationarity of performance, and mutual influences of cases over time in detailed and aggregated form. In the following, we show that aggregate PPM can be predicted directly from features of the PS as the richness of the PS directly encodes inter-case performance characteristics over time. Fig. 1 illustrates a simplified PS that shows that the workload and performance of all cases at step d in the future depends on the recent throughput in steps a-c *together*. This allows us to reduce the IC3P problem to an ML problem over features of the PS. To achieve this result, we introduce the notion of "feature channels" to capture different process and performance perspectives in the PS and to describe different PPIs over time and over process steps.

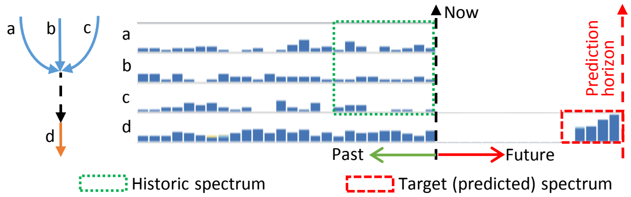

Fig. 1. The load peak on the X-ray baggage screening machine of a BHS (process step d) can be computed through the recent historic load on the check-in counters (process steps a-c) rather than individual properties of cases.

We provide a general formal formulation of the inter-case performance prediction problem over the multi-channel PS; and a methodology for formulating problem instances (especially feature selection and reduction) and solving the problem (using standard ML for model training). Our evaluation on simulated and real-life data demonstrates the feasibility of our approach and that prediction of aggregated PPM using PS-based features outperforms prediction using case similarity-based features [7].

The remainder of this paper is structured as follows. We discussed work related to PPM of MHS and BP in Sect. II. We recall the PS and introduce the multi-channel PS in Sect. III. We formally define the generic problem of the IC3P in Sect. IV and propose the methodology for solving it in Sect. V. We report on our evaluation on synthetic and real-life event logs in Sect. VI and discuss our findings and future work in Sect. VII.

## II. Related Work

For BP, the remaining processing time for a case can be predicted by regression models [8] or by decorating a transition system with remaining time [9], prior trace clustering improves the prediction [10]. In [11], a Naive Bayes classifier predicts the future path of a single running case and a regression model predicts the transition durations on this path. The likelihood of future activities can be predicted using Markovian models [12], but without providing any time predictions. Completion time of the next activity can be predicted by training an LSTM neural network [13], or by learning process models with arbitrary probability density functions for time delays through non-parametric regression from event logs [14] that can also be used for learning simulation models to predict performance [15], [16]. Competing for shared resources can be taken into account through simulation models or with queuing models [17]. Using only features of a single case, these models cannot predict PPIs for non-isolated cases. Estimating an aggregate PPI through the outcome of individual cases [18] cannot be used for non-mandatory outcome of non-isolated cases. Prediction of the remaining time for a single case in processes with non-isolated cases is addressed in [7], intra-case features of a running case of interest are coupled with inter-case features of concurrently running cases, “close” to the case of interest in terms of control-flow and temporal distances. However, in processes with tightly coupled dynamics such as MHS, cases influence each other, e.g., congestions propagate through the system and resource problems affect groups of cases, impacting the performance. The PS-based approach in this paper specifically captures this dynamics. Since [7] is the current state-of-the-art approach for inter-case feature encoding, we use it as a baseline.

Among MHS, BHS are studied extensively. In the BHS domain, relationships between some bag- and system-related properties can be learned by feedforward NN models [4], but the results reported as just acceptable, even for a fully controllable environment of a simulation model. A risk of baggage mishandling can be predicted with an aggregated probabilistic flow graph as a function of travel durations between system locations [1], while dynamic routing is not supported. Problem-oriented *simulation models* allow identifying of bottlenecks and critical operations for inbound baggage handling [19]; learning dependencies between security policies and time characteristics of manual baggage screening [3]. In [2] an overview of various simulation-based performance prediction techniques for baggage screening is provided. While these simulation models are precise, their design requires in-depth knowledge of a system design and proved to be time-consuming.

Our work contributes to the problem of predicting aggregate PPIs for processes with *non-isolated* cases that *influence each other*. We capture inter-case dependencies by leveraging the performance spectrum that we recall next, and learn unknown system behavior from performance-related features of the performance spectrum, thereby extending the application of non-simulation-based approaches of PPM to MHS.

## III. Performance Spectrum

We first establish some basic notations for events and logs, recall the idea of the Performance Spectrum (PS) and revise the definitions of [6] to provide “elementary” PS building blocks from which we construct a *multi-channel* PS for performance prediction.

Let $A$ be a set of *event classifiers*; $A$ is usually the set of activity names, but it may also be the set of resource names, or locations. Let $T$ be the set of time durations and time stamps, e.g., the rational or real numbers. Let $\mathcal{E}$ be the *universe of events* with *attributes*, and let $AN$ be a set of attribute names. For any $e \in \mathcal{E}, n \in AN$, $\#_n(e)$ is the value of attribute $n$ for event $e$ ($\#_n(e) = \perp$ if attribute $n$ is undefined for $e$). Each event has mandatory attributes *time*, $\#_{time}(e) \in T$ and *act*, $\#_{act}(e) \in A$. As a short-hand, we write $e^{(a,t)}$ to indicate that event $e$ has $\#_{act}(e) = a$ and $\#_{time}(e) = t$. Let $\mathcal{Z}$ be the *universe of cases* with attributes. For any $z \in \mathcal{Z}, n \in AN$, $\#_n(z)$ is the value of attribute $n$ for case $z$. Each case has a mandatory attribute *trace*, $\#_{trace}(z)$, defining a finite sequence of events $\#_{trace}(z) = \sigma \in \mathcal{E}^*$. For $\sigma = \langle e_1, \ldots, e_n \rangle$, we write $|\sigma| = n$ and $\sigma_i = e_i, i = 1, \ldots, n$. An event log is a set of cases $L \subseteq \mathcal{Z}$ where no two traces share an event.

The *Performance Spectrum* is a *data structure* introduced in [6] to describe the performance of process steps over time. We first recall the idea and then adopt it for performance prediction. We call $(a, b) \in A \times A$ a *process segment* describing a step from activity $a$ to activity $b$, hand-over of work from resource $a$ to $b$ or the movement of goods from location $a$ to $b$. Each occurrence of a segment $(a, b)$ in a trace $\langle \ldots, e_i^{(a,t_a)}, e_{i+1}^{(b,t_b)}, \ldots \rangle$ allows to measure the time $t_b - t_a$ between occurrences of a and b. A histogram $H = H(a, b, L) \in \mathbb{B}(T)$ describes how often all the *time differences* $t_b - t_a$ between $a$ and $b$ have been observed in $L$. In contrast, the *performance spectrum* $\mathbb{S}(a, b, L)$ collects the actual *time intervals* $(t_a, t_b)$ observed in $L$. Fig. 2(a) shows the so-called *detailed PS* for the segment $(a, b)$: each dot along the $a$-axis marks an occurrence of an $a$-event, correspondingly $b$-events are shown on the $b$-axis. The diagonal line $(a_1, b_1)$ describes one *occurrence* of the segment from event $e_i^{(a,t_a)}$ to $e_{i+1}^{(b,t_b)}$ in the same trace, e.g., the movement of a bag between two locations. Different lengths of segment occurrences indicate performance differences for different cases; changing density of segment occurrences indicates changing workload on the segment over time.

To allow *computing* with the visually evident performance-related features of Fig. 2(a), the PS may also provide *classification* and *aggregation* of the occurrences of a segment. In the following, we revise these definitions to allow defining basic building blocks for process segment, classification, and time

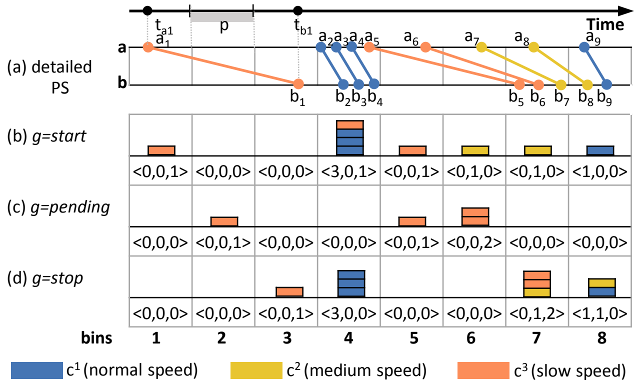

Fig. 2. In the detailed PS (a) the color-coded lines show cases with different speed classes, while the aggregated PS with various grouping (b-d) capture various performance aspects of case handling for each time window.

interval of interest. In [6], occurrences $\langle \dots, e_i^{(a,t_a)}, e_{i+1}^{(b,t_b)}, \dots \rangle$ of $(a,b)$ were classified wrt. duration, e.g., the actual duration $\Delta t = t_b - t_a$ or whether $\Delta t$ is in the 25% quartile of the histogram $H(a,b,L)$. Here, to enable general performance prediction, we assume a function $\mathbb{C}$ that maps two events $e_i, e_{i+1}$ to a *performance class* $\mathbb{C}(e_i, e_{i+1}, L) = c \in C$ considering *any* properties of $e_i, e_{i+1}$ and the log $L$ in which they occur. We call $\mathbb{C} : \mathcal{E} \times \mathcal{E} \times 2^{\mathcal{Z}} \to C$ a *performance classifier* for $C$. Examples are the duration from $e_i^{(a,t_a)}$ to $e_{i+1}^{(b,t_b)}$, remaining time until case completion since $e_{i+1}$, or whether a material had to be routed from $a$ to $b$ because an alternative route from $a$ was blocked; scalar values may be abstracted to categories.

If the performance classes $C$ are *finite*, the detailed PS of a segment $(a,b)$ can be *aggregated* over "bins" of a chosen, fixed duration $p$, called *bin size*. For each bin $b_j$ and class $c \in C$ we count how many occurrences of segment $(a,b)$ of class $c$ occur "during" $b_j$. As Fig. 2 illustrates, we may choose to count the segments that *start* during $b_j$ ($e_i$ is in $b_j$ but $e_{i+1}$ is not) (b), *stop* during $b_j$ (d) or are *pending* (c). For example, segment occurrence $a_1 b_1$ has class $c^3$ (color pink), starts in bin 1, is pending in bin 2 and ends in bin 3. Suppose we chose to group on *start*, then, we aggregate this information into a vector $\langle v_j^1, v_j^2, v_j^3 \rangle$ where, say, $v_j^3$ counts the number of segment occurrences of class $c^3$ that occurred during bin $j$. Figure 3(b-d) shows the aggregation vectors for each grouping and each bin and their visualization as stacked barcharts. Def. 1 and 2 formalizes these for a single segment, classifier, and bin. We canonically lift them to multiple classes, segments, and bins afterward.

**Definition 1** (Detailed performance spectrum (Detailed PS))**.** *Let $\mathbb{C}$ be a performance classifier for $C$. Let $L$ be a log and $(a,b)$ be a segment. The* detailed PS *of $(a,b)$ in $L$ wrt. $\mathbb{C}$ is the bag $\mathbb{S}_L((a,b),\mathbb{C}) = [(t_a,t_b,c) \mid \langle \dots e_i^{(a,t_a)}, e_{i+1}^{(b,t_b)} \dots \rangle = \#_{trace}(z), z \in L, 1 \le i < |\#_{trace}(z)|, c = \mathbb{C}(e_i^{(a,t_a)}, e_{i+1}^{(b,t_b)}, L)] \in \mathbb{B}(T \times T \times C)$.*

In Fig. 2(a) elements $(t_a, t_b, c)$ of the PS are visualized as lines that start at time moments $t_a, t_b$ on axes a,b, class c is indicated by color.

**Definition 2** (Aggregation of PS)**.** *Let $\mathbb{C}$ be a performance classifier for finite classes $C = \{c_1, \dots, c_k\}$. Let $S = \mathbb{S}_L((a,b),\mathbb{C})$ be the detailed PS of a segment $(a,b)$ in a log $L$. Let $p \in T$ be a duration we call the* bin size *and let $g \in \{start, stop, pending\}$. The occurrences of $(a,b)$ in bin $j \in \mathbb{N}$ (of length $p$) regarding grouping $g$ is the multiset $b_j$ such that:*

- *$b_j = [(t_a,t_b,c) \in S \mid j \cdot p \le t_a < (j+1) \cdot p]$ if $g = start$,*
- *$b_j = [(t_a,t_b,c) \in S \mid j \cdot p \le t_b < (j+1) \cdot p]$ if $g = stop$, and*
- *$b_j = [(t_a,t_b,c) \in S \mid j \cdot p > t_a \wedge t_b \ge (j+1) \cdot p]$ if $g = pending$ (the segment starts before the start of the bin, and ends after (or at) the end of the bin).*

*The* aggregation *of $S$ over bin $j$ and grouping $g$ is the vector $v_j = \langle v_j^1, \dots, v_j^k \rangle \in \mathbb{N}^k$ counting how often performance class $c^i$ occurred in bin $v_j$: $v_j^i = |[(t_a,t_b,c^i) \mid (t_a,t_b,c^i) \in b_j]|$. Let $\mathbb{S}_L((a,b),\mathbb{C},g,p,j) = v_j$.*

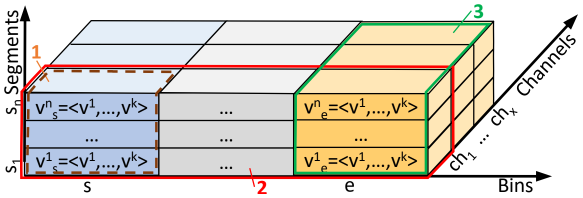
Segments
$s_n$
$s_1$
1
2
3
$v^n_s$=<$v^1$,...,$v^k$>
...
$v^1_s$=<$v^1$,...,$v^k$>
...
$v^n_e$=<$v^1$,...,$v^k$>
$v^1_e$=<$v^1$,...,$v^k$>
s
e
Bins
Channels
$ch_1$ ... $ch_x$

Fig. 3. The multi-channel Performance Spectrum.

For example, in Fig. 2(d) the aggregation of the PS for segment $(a,b)$ over bin 7 and grouping $g = stop$ is vector $\langle 0,1,2 \rangle$ which counts ends of all segment occurrences in this bin: zero for class $c^1$, one for $c^2$ ($b_7$) and two for $c^3$ (points $b_{5-6}$).

The aggregation of a detailed PS into a bin has 3 main dimensions: (1) the segment $(a,b)$, (2) the parameters describing the bins, i.e., the classification $\mathbb{C}$, the grouping $g$, and the period $p$, and (3) the bin number $j$. To simplify notation, we call the bin parameters $ch = (\mathbb{C}, g, p)$ a *PS channel*, and write $\mathbb{S}_L((a,b), ch, j) = v_j$ for the aggregation vector of Def. 2.

We now show how a bin $\mathbb{S}_L((a,b), ch, j)$ of the aggregated PS is the basic building block to formulate various performance prediction problems. Consider Fig. 3: each bin of the aggregated PS can be placed in a 3-dimensional space defined by a series of segments $SEG = \langle (a_1,b_1), \dots, (a_n,b_n) \rangle$, a series of channels $CH = \langle ch_1, \dots ch_x \rangle$, and a time interval of bins $[s,e] = \langle s, s+1, ..., e \rangle$ of interest. We use slicing and dicing in this 3d-data structure to define our prediction tasks. Adopting notation from algebra software, we let the arguments of $\mathbb{S}_L(\cdot,\cdot,\cdot)$ range over sequences of segments, channels, and bin numbers to denote rows, columns, matrices, and cubes of bins along those dimensions. Let $SEG = \langle (a_1,b_1), \dots, (a_n,b_n) \rangle$ be a sequence of segments, $CH = \langle ch_1, \dots, ch_x \rangle$ be a sequence of PS channels (of identical period $p$), $ch \in CH$, and $[s,e] = \langle s, s+1, \dots, e \rangle$ a sequence of bin numbers, $j \in [s,e]$. We write $\mathbb{S}_L(SEG, ch, j)$ for the column vector $\langle \mathbb{S}_L((a_1,b_1), ch, j), \dots, \mathbb{S}_L((a_n,b_n), ch, j) \rangle^\top$. Note that this vector consists of vectors $v_j^i$ for each segment $(a_i, b_i)$ and bin $j$. In Fig. 3 such a column vector corresponds to a column of blocks of size $n \times 1 \times 1$, e.g. area (1).

We write $\mathbb{S}_L(SEG, ch, [s,e])$ for the row vector $\langle \mathbb{S}_L(SEG, ch, s), \dots, \mathbb{S}_L(SEG, ch, e) \rangle$. Note that each $j^{th}$ entry of this vector corresponds to a column vector $\mathbb{S}_L(SEG, ch, j)$. In Fig. 3 $\mathbb{S}_L(SEG, ch, [s,e])$ corresponds to a

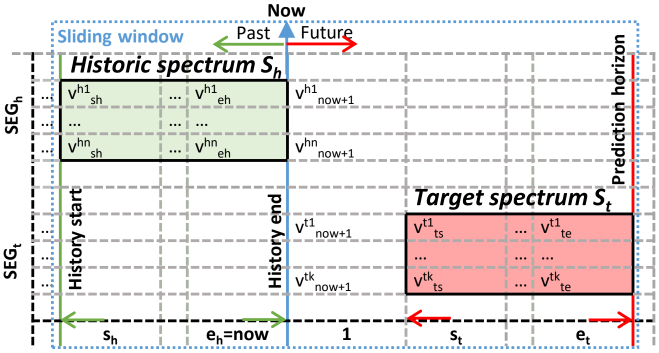

Fig. 4. The configuration of the historical and target spectra “around” the current time of the sliding window: the historic spectrum in the past is used to compute the target spectrum in the future.

frontal ’slice’ of size $n \times 1 \times (e - s + 1)$ for channel $ch$ , e.g. area (2). We use this matrix to *visualize* the performance spectrum of the segments $SEG$ over the time period $[s, e]$ in a single channel $ch$. For example, Fig. 2(b) visualizes one row of such a matrix and Fig. 3 visualizes an entire matrix. We also this visualization of the frontal slice for feature selection in Sect. V. Performance prediction requires considering information from *multiple* channels during the same time period. We write $\mathbb{S}_L(SEG, CH, j)$ for the column vector $\langle \mathbb{S}_L(SEG, ch_1, j), \ldots, \mathbb{S}_L(SEG, ch_x, j)\rangle^\top$. Note that each $r^{th}$ entry of this vector corresponds to a column vector $\mathbb{S}_L(SEG, ch_r, j)$. In Fig. 3 $\mathbb{S}_L(SEG, CH, j)$ corresponds to a vertical ’slice’ of a single bin columns of size $n \times x \times 1$, e.g. area (3), where it is shown as a matrix over segments and channels. Note that the order of segments and channels is arbitrary but fixed to allow implicit encoding of features, whereas the order of bins is determined by time.

We write $\mathbb{S}_L(SEG, CH, [s, e])$ for the row vector $\langle \mathbb{S}_L(SEG, CH, s), \ldots, \mathbb{S}_L(SEG, CH, e)\rangle$, which correponds to the whole cube in Fig. 3. Note that this vector is a matrix with columns corresponding to bins, i.e. time, and rows corresponding to segments and channels. Such a structure allows to slice and dice it in various ways. Row vectors can be used for visual analytics, columns vectors of various bin intervals can serve for extracting independent and dependent variables for model training. Aggregation along the bin and segment axes allows for feature space reduction. We call $\mathbb{S}_L(SEG, CH, [s, e])$ the *multi-channel* performance spectrum of $L$ over segments $SEG$, channels $CH$, and period $[s, e]$.

## IV. Problem Statement

The Inter-Case Performance Prediction Problem (IC3P) is to obtain a model for predicting the performance of multiple cases together, in a specific part of the process, within a particular prediction window. In this section, we show how the features of the multi-channel PS of Sect. III allow formulating precise IC3P instances.

### A. Problem Formulation with Performance Spectrum

Along the dimensions of the multi-channel PS, the IC3P is to predict the performance characteristics for segments of interest for a time-interval in the future (the *target* spectrum), based on the performance of relevant segments during a recent time interval (the *historic* spectrum). An estimate for a specific PPI can then be derived by aggregating performance-related features of the target spectrum. In Fig. 4 a schematic configuration of such a problem is shown for one channel of a multi-channel PS. The historic spectrum is specified by a sequence $SEG_h = \langle s^1_h, \ldots, s^n_h\rangle$ of segments and a bin interval $[s_h, e_h]$ where $s_h < e_h \leq 0$ define *offsets* from the current bin *now* in the sliding window. The target spectrum is given by segments $SEG_t$ and a bin interval $[s_t, e_t], 0 \geq s_t > e_t$ with offsets into the future. Index $e_t$ defines the prediction horizon (PH). Both historic and target spectrum are defined over the same sequence $CH$ of channels. This allows formulating IC3P as a regression problem, using historic and target spectra as a source of independent and dependent variables over common time parameter $T$, formalized in (1):

$$\begin{aligned} &\mathbb{S}_L(SEG_t, CH, [s_t + T, e_t + T]) = \\ &f(\mathbb{S}_L(SEG_h, CH, [s_h + T, e_h + T])) + R, \end{aligned} \quad (1)$$

or $S_{L,t}(T) = f(S_{L,h}(T)) + R$ for short. Function $f$ *predicts* values of the target spectrum, and $R$ is a residual, i.e. the deviation between observed and predicted values. To learn $f$, we use the sliding window method [20] for selecting $w$ samples $(S_{L,h}(T_i), S_{L,t}(T_i))$ of historic and target spectrum for times $T_1, \ldots, T_w$, and apply a ML method to learn $f$ from these samples. By comparing the actual values $y = \langle S_{L,t}(T_1), \ldots, S_{L,t}(T_w)\rangle$ in the target spectrum with the values predicted by learned function $f$, $y' = \langle f(S_{L,h}(T_1)), \ldots, f(S_{L,h}(T_w))\rangle$, we can estimate the prediction error $R$ in $f$ by a function $error(y, y') \in \mathbb{R}$. In general, a target spectrum does not contain the target PPI directly, but contains performance-related features sufficient to compute it. For that, we define the following function:

$$ppi(T) = g(\mathbb{S}_L(SEG, CH, [s_t + T, e_t + T])) + \varepsilon, \quad (2)$$

where error $\varepsilon = ppi(T) - ppi'(T)$ and $ppi'(T)$ is the predicted target PPI observed over interval $[s_t, e_t]$.

### B. Examples of Problem Instances

We now instantiate the generic problem formulation from (1) for concrete real-life performance prediction problems of a major European airport baggage handling system (BHS). A fragment of its simplified material flow diagram is shown in Fig. 5. We first consider the process from check-in until screening. Bags enter the system via one of several dozen check-in counters $a^1_1$-$a^m_n$ and then move via conveyor belts to one of two pre-sorter loops P1,P2 where each bag has to go to the X-ray baggage screening machines, e.g. entering via $(E_1, S_1)$ and leaving via $(S_2, X_1)$. For operational support, the main concern is to keep the BHS performance steady at some desired level. In particular, the workload in a processing step or system part may not exceed its capacity, as this otherwise leads to long queues or stalling of sorting loops. Here, workload *prediction* is central for proactive management. One concrete problem (*PI1*) is to predict the load (in bags per minute) at the X-ray baggage screening machines (SM) on $t_{PH} = 4$ minutes

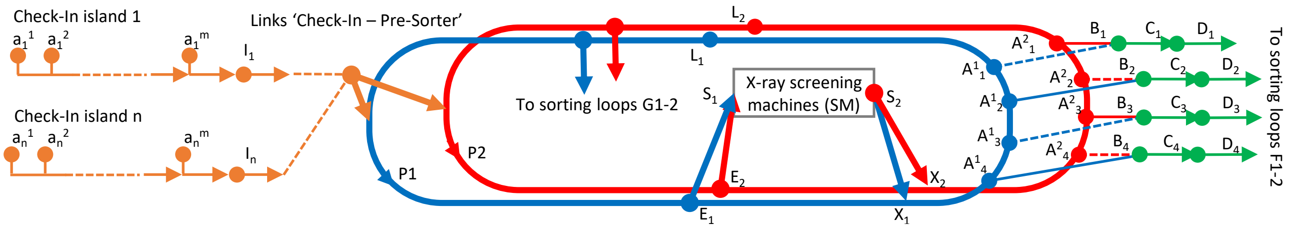

Fig. 5. Check-in and pre-sorting areas of a Baggage Handling System.

in advance for P1. In Fig. 5 this load corresponds to the load on segment $SEG_{t,PI1} = \langle (E_1, S_1) \rangle$.

To express this problem in terms of (1), we define the target and historic spectra. First, to represent the load, we define a PS channel with a single class (load does not distinguish different classes), grouping *start* and 1-minute bin: $ch_{PI1} = \langle (\mathbb{C}_{PI1}, start, 1) \rangle$, where $\mathbb{C}_{PI1}$ returns zero for any segment occurrence. For log $L$, the target spectrum for *PI1* is $S_{PI1}(T) = \mathbb{S}_L(SEG_{t,PI1}, ch_{PI1}, [t_{PH} + T, t_{PH} + T])$. The predicted load is the sum of all its values, in bags per minute. Then we make the following hypothesis: the load depends on the average load of the check-in counters in 1-3 minutes before *now*. To capture that, we include the check-in segments to the historic spectrum: $SEG_{h,PI1} = \langle (a_1^1, I_1), \ldots, (a_1^m, I_1), \ldots, (a_n^1, I_n), \ldots, (a_n^m, I_n) \rangle$ and time-interval $[s_h^{PI1}, e_h^{PI1}]$ as $[-3, -1]$. This leads to the following regression problem derived from (1):

$$S_{PI1}(T) = f_{PI1}(\mathbb{S}_L(SEG_{h,PI1}, ch_{PI1}, [T-3, T-1]) + R. \quad (3)$$

The target PPI is defined as $ppi_{PI1}(T) = S_{PI1}(T)_1$, where the index means the aggregate of the fist performance class in $\mathbb{C}_{PI1}$.

Another concern of BHS operational support is predicting the risk that baggage being late for a flight; we now instantiate (1) for this problem (*PI2*). The second part of the process in Fig. 5 moves bags from the screening machines to sorting loops F1,F2 (exit the pre-sorter P1 and P2 via $A_1^1$-$A_2^4$, $B_1^1$-$B_2^4$). It may happen that, for instance, a bag on P1 that has to go to F1, cannot be diverted onto any of the conveyors $(A_i^1, B_i)$ because these are unavailable (e.g., due to high load on all $(C_i, D_i)$. In this case, the bag will be looping on P1 until it can be diverted successfully. Each round increases the bag's *estimated time to destination* (EST) $t_{est}$ by the loop duration $t_P$. If the new estimate $t'_{est} = t_{est} + t_p$ exceeds the deadline when the bag has to arrive at its destination to reach the flight, the bag is expected to be late and correcting actions, e.g. making the bag priority higher, can be undertaken.

So, to predict such late bags, it is sufficient to predict extra recirculation due to unavailability of diverts $A_1^1$-$A_2^4$. We formulate *PI2* as a problem of predicting such re-circulation for P1. On P1, any bag traveling the segment $(A_1^1, L_1)$ is re-circulating (as it could not be diverted to F1,F2), thus the segments of the target spectrum are $SEG_t = \langle (A_1^1, L_1) \rangle$. Selecting $t_{PH} = 60 seconds$, duration-based classifier $\mathbb{C}_{PI2}$ (whether $t = t_b - t_a$ is in the 25%-quartile of the histogram $H(a, b, L)$) and $chs_{PI2} = \langle (\mathbb{C}_{PI1}, start, 30 seconds), (\mathbb{C}_{PI2}, pending, 30 seconds) \rangle$, we make a hypothesis, that the target spectrum depends on the load and delays of $SEG_{h,PI2} = \langle (S_2, X_1), (S_2, X_2), (A_i^1, B_i), (A_i^2, B_i), (B_i, C_i), (C_i, D_i) \mid i = 1, \ldots, 4 \rangle$ for two bins before *now*. We predict the target spectrum $S_{PI2}(T) = \mathbb{S}_L(SEG_{t,PI2}, chs_{PI2}, [T + 1, T + 1])$ as follows:

$$S_{PI2}(T) = f_{PI2}(\mathbb{S}_L(SEG_{h,PI2}, chs_{PI2}, [T-2, T-1]) + R. \quad (4)$$

The PPI is defined as $ppi_{PI2}(T) = \mathbb{S}_L(SEG_{t,PI2}, chs_1, [T + 1, T + 1])_1 + \varepsilon$, i.e. we select channel with grouping *start* and the first performance class in $\mathbb{C}_{PI2}$.

## V. Approach

In Sect. IV, we showed that prediction of aggregate performance measures for non-isolated cased can be expressed as a generic regression problem over the performance spectrum. In this section, we present a general methodology on formulating concrete problem instances and how to solve them using a standard machine-learning pipeline. Fig. 6 illustrates the overall methodology.

The main challenge is to correctly select the features for defining the historic and the target spectra. In the following, we summarize some lessons learned from our experiments that we discuss in Sect. VI.

### A. Methodology

In Step 1, *target segments* for the problem, i.e. the segments, which performance-related features are sufficient for computing the target PPI, are identified and located in the model. In Step 2, the target segments are considered for aggregation. MHS equipment is usually redundant, to provide high availability and fault tolerance of the whole system. For example, several baggage screening machines, working in parallel, are usually grouped in a cluster with symmetrical layout and some load balancing policy. A set of similar-purpose segments $(a_1, b_1), \ldots, (a_n, b_n)$ can be aggregated into a new aggregated segment $(a^*, b^*)$ by relabeling $a_i \mapsto a^*$, $b_i \mapsto b^*$ prior to computing the PS.

Similar-purpose segments within such clusters can be aggregated in the PS to reduce the feature vector along the *SEG* dimension (see Fig. 3). During Step 3 we define PS channels that contain features, required for computing the target PPI, by choosing a common granularity (period $p$), classifiers and groupings. This step is specific to the problem. For example, to compute load on a segment, a combination of grouping *start* with a constant (single-value) classifier may be sufficient, as for *PI1*, while for counting performance outliers

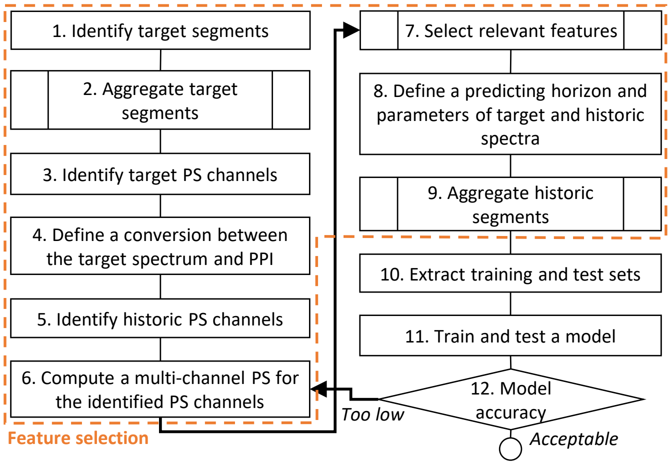
1. Identify target segments
2. Aggregate target segments
3. Identify target PS channels
4. Define a conversion between the target spectrum and PPI
5. Identify historic PS channels
6. Compute a multi-channel PS for the identified PS channels
Feature selection
7. Select relevant features
8. Define a predicting horizon and parameters of target and historic spectra
9. Aggregate historic segments
10. Extract training and test sets
11. Train and test a model
12. Model accuracy
Too low
Acceptable

Fig. 6. The methodology on formulating problem instances of the inter-case performance prediction problem.

another grouping *pending* with a duration-based classifier is required. Then in Step 4, given the identified target spectrum parameters, a concrete function $g$ (2) is defined. During Step 5, historic PS channels are identified to take into account more features for estimating the target spectrum. While period $p$ is common for all the PS channels, particular classifiers and groupings for these channels depend on the problem and system. Using domain knowledge and/or performance analysis results of earlier iterations, additional PS channels can be included into the historic PS channels vector. Next in Step 6, a multi-channel PS is computed for the identified PS channels.

In Step 7 we should answer the following question: which features of the multi-channel PS influence the target spectrum and should be included in the historic spectrum? For that historic segments and time boundaries of the historic spectrum should be defined. We suggest using the computed multi-channel PS as a visual analytics technique for feature selection according to the following guideline. We formulate the following high-level guideline, which describes the main steps of such analysis. First, a *Segment Group of Candidates* (SGC) to the historic spectrum is identified. The focus is usually on segments that are in several steps upstream and downstream the target segments. Additionally, all segments that *a priori* affect the target spectrum (according to domain knowledge) are included. Afterward, the correlation between the target spectrum features and features of segments in the SGC can be determined. For example, in Fig. 7 an interval of higher load on segment $s_1$ causes a higher load on target segment $s_t$, so this segment should be included into the historic spectrum. Finally, the following questions should be answered. Which segments of the SGC influence the target spectrum? What is an average delay of affecting the target spectrum ($\Delta_1$ in Fig. 7)? What is the time interval of the historic spectrum that should be used to estimate the target spectrum ($l_1$ in Fig. 7)?

In Step 8, the Prediction Horizon (PH), i.e., the moment of prediction, is chosen in light of the dynamics from the segments and bins in the historic spectrum that dominate the target spectrum. After this step completion, all required parameters of the target and historic spectra are defined: the segments names, start and stop indices. Similarly to Step 2, in Step 9 the historic segments are considered for aggregation.

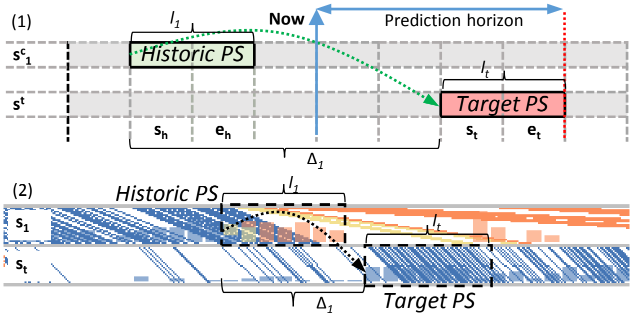

Fig. 7. Visual analytics over the PS: segment $s_1$ influences target segment $s_t$.

In Steps 10-11 a standard ML pipeline is exploited for model training. The multi-channel PS, built in Step 6, is used directly for the extraction of the training and test sets. Using the sliding window technique [20], the historic and target sets are instantiated for each bin of the multi-channel PS, using the parameters identified in the previous steps, and stored as a sample of the training or test set for the consecutive model training. After a model is configured, trained and tested, a decision on the model accuracy is made. If it is lower than required, more iterations can be done, e.g. to change the non-target PS channels, aggregation rules, selected features, PH and model configuration in order to improve the model accuracy.

In Sect. VI we will apply this approach on *PI1* and *PI2*.

## VI. Evaluation

We extended the interactive ProM plug-in 'Performance Spectrum Miner' with the multi-channel PS and scripts for training models in the PyTorch ML framework[1]. We demonstrated the feasibility of our approach and compared it to the current state-of-the-art approach [7] by training models for *PI1* and *PI2* of Sect. IV-B (details of data are in Sect. VI-A and VI-B). For the experiments we applied the training-validation-test approach, using 20% of the data for testing. The remaining part was randomly shuffled and used for model training and 5-fold cross-validation in proportion 4:1. For model learning, we evaluated three approaches. (1) For our approach, we extracted PS-based features as discussed in Sect. IV and trained Logistic Regression (LR) and Feedforward (FF) Neural Network (NN) models that predict the expected load in the target segments directly. (2) As a baseline capturing both intra- and inter-case dependencies, we chose [7][1]. As it only predicts PPIs for individual cases, we had to adopt it for the aggregate PPI as follows. First, we trained a model for predicting the time between *last events of trace prefixes* that end with occurrences of historic spectrum segments and *starts of target segments*, using LR and FF NN models. For each prefix the learned model predicts when it will reach the "target". By aggregating how many cases are predicted to reach the "target" bin, we can estimate the expected load. Because such an aggregation can be done only for cases that eventually

[1] the simulation event log, ProM plugin, PyTorch script and source code of [7] are available at https://github.com/processmining-in-logistics/psm/tree/ppm

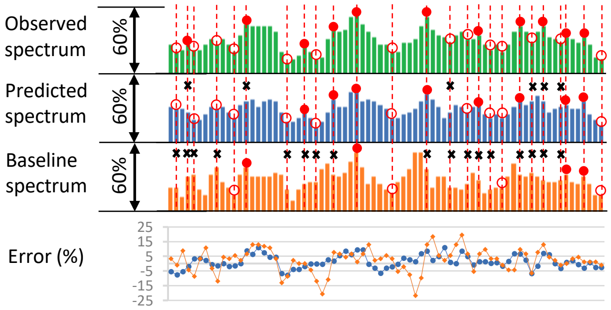

Fig. 8. The real, predicted (PS-based approach (1)) and baseline (approach (2)) load of the baggage screening machines, in % of the maximal load (top): each bin represents the load for one minute, filled and blank circles show matched peaks and dips, while the X's show mismatches. The residuals of the predicted load in % of max. load per bin (bottom): the baseline (orange) shows greater deviations than the PS-based model (blue).

reach target segments, it assumes beforehand knowledge about future paths of cases. This assumption holds for historic data on the model training stage, but does not hold for real MHS on the prediction stage. For example, in the real BHS, considered in Sect. VI-B, bags can be routed to P2 (see Fig. 5 and never reach the target segment $(E_1, S_1)$ (PI1), or bags from P1,P2 can be sent to sorters G1,G2 instead of F1,F2, thereby being outside the scope of the re-circulation problem (PI2). (3) As a naive baseline, we chose an average value of dependent variables, observed in a time interval $[s_h, e_h]$, corresponding to the historic spectrum. To measure errors (see Sect. IV), we computed *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE) and *R squared*, which is meaningful for linear models. Additionally, we did meticulous residual diagnostic of predictions for test sets. Models were trained on a server with 40 CPUs, six GPUs and 400 GB memory.

### A. Simulation Model of a Baggage Handling System

To generate an event log for addressing *PI1*, we designed a simulation model of a simple BHS, comprising a typical BHS layout: conveyors, a sorting loop, a baggage screening machine, divert and merge units. As a load, a check-in scenario with normally distributed distances between bags was replayed to generate an event log with 134.000 events and 11.518 cases for 84 operating hours. Events were recorded when bags passed through various locations in the system. The resulting training and test sets have 15 feature variables and 15.000 samples[1], on which we applied approaches (1-3). Table I (a) shows the resulting measures. Our PS-based models show two times smaller errors RMSE and MAE than approach (2). The PS-based LR model has a greater and closer to 1.0 R squared measure than the LR model of approach (2), i.e. it explains significantly more variable variations.

### B. Baggage Handling System of a Major European Airport

In this experiment, we addressed *PI1* and *PI2* for a Vanderlande-built BHS of a major European airport. In the event log, each case corresponds to one bag, events are recorded when bags pass sensors on conveyors, and activity names describe locations of sensors in the system. Events are recorded only when a bag is diverted to another conveyor, so information about the bag locations is significantly incomplete. For one day of operations, an event log contains on average 850 activities, 25.000-50.000 cases and 1-2 million events. The entire log contained 148 million events for 120 consecutive days. Events recorded in non-operating night hours were excluded from the log. For the test set, days of different months and days of week were selected.

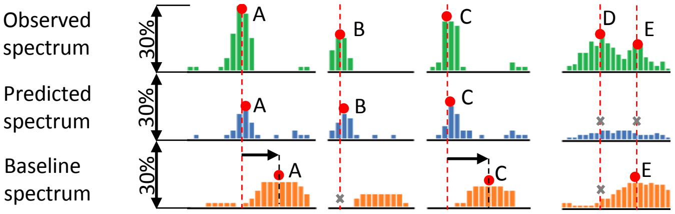

Fig. 9. The peaks of re-circulation within 30-second bins (in % of the max. load): the FF NN model predicted peaks A, B, C in the correct bins, whiles the baseline predicted them with the significant delay as a result of auto-correlation. Part of peaks (e.g. D and E) were not predicted by the model.

| Experiment | Approach | Model | R squared | RMSE | MAE |
|---|---|---|---|---|---|
| | (1) | LR | 0.82 | 12.1 | 8.2 |
| (a) | (1) | FF NN | 0.84* | 11.6 | 7.7 |
| BHS sim. model | (2) | LR | 0.35 | 23.2 | 15.3 |
| PI1 | (2) | FF NN | 0.46* | 21.2 | 16.2 |
| | (3) | - | - | 35.8 | 23.2 |
| (b) | (1) | LR | 0.74 | 7.0 | 5.0 |
| Real BHS | (1) | FF NN | 0.75* | 6.9 | 4.8 |
| PI1 | (2) | LR | 0.38 | 10.8 | 7.9 |
| | (2) | FF NN | 0.69* | 7.6 | 5.3 |
| | (3) | - | - | 11.0 | 7.9 |
| (c) | (1) | LR | 0.05 | 3.0 | 1.8 |
| Real BHS | (1) | FF NN | 0.45* | 2.4 | 1.3 |
| PI2 | (3) | - | - | 3.0 | 1.7 |

TABLE I
Model error measures for PS-based (1), [7]-based (2) and naive (3) approaches. RMSE and MAE are in % of max. load (a,b) and re-circulation (c).
**R squared values for FF NN are provided for the sake of completeness.*

First, we addressed *PI1*. Following the approach in Sect. V, for approach (1) we trained an LR model and a two-layer FF NN on a dataset with 68 features and 108.000 samples. As shown in Tab. I (b), both PS-based models have smaller RMSE and MAE errors than the baseline models, and the R squared measure of the LR model is also good. Fig. 8 shows how the LR model correctly predicts peaks and dips in the load of the scanning machines compared to the recorded data, suggesting that the model is adequate for the workload prediction. The LR model is preferred for the sake of simplicity.

Finally we addressed *PI2*. Again, for approach (1) we trained an LR model and a four-layer FF NN on a dataset with 148 features and 216.000 samples (two times more than for the previous experiment because of the shorter bin duration), using approach (3) as a baseline. Tab. I (c) shows almost zero R squared of the LR model, that indicates incapability to explain variable variations, i.e. the model cannot predict infrequent re-circulation peaks, while RMSE and MAE of FF NN models are smaller than corresponding values of the baseline. Fig. 9 shows that the FF NN model correctly predicts moments of peaks in re-circulation, but consistently underestimates its actual amount, while the baseline demonstrates auto-correlation.

Despite incompleteness of the log, the sound PS-based models, trained during the experiments, demonstrated the feasibility of the suggested approach for PPM problems of the real BHS as well as for the simulation model.

## VII. Conclusion

In this paper, we studied the problem of forecasting performance of non-stationary processes where cases influence each other through shared resources. We showed that the performance spectrum (PS) [6] derived from the event log of a process allows to model a variety of process performance features over time, capturing also inter-case dependencies. Specifically, we provided a basic building block defined over the three basic dimensions of process step, performance measure, and time interval. We showed that combining multiple such blocks of features in a *multi-channel* PS along the three dimensions allows formulating a large class of performance prediction problems as a regression problem. We proposed a methodology of solving this problem as a ML task, using the historical and target spectrum features as independent and dependent variables. The methodology includes the approach for feature selection, based on visual analytics of individual channels within the multi-channel PS, and process model-based aggregation of process segments for feature dimensionality reduction. We provided examples of real-life problem instances for a BHS and evaluation of our approach by training sound models for solving these problem instances on the real event log of a major European airport BHS. We demonstrated feasibility of our approach and compared it to the current state-of-the-art approaches, e.g., [7]. The experiments showed that our PS-based linear model outperforms more complex NN model of [7]-based approach, and the PS-based NN model outperforms the naive baseline for the problem instance where [7] is not applicable due to the optionality of the target process step. This work has several limitations. First, although supported by a methodology, feature selection and reduction requires domain knowledge and expertise. We expect that including a formal model of the process may help in engineering features from the performance spectrum. Second, while our models are technically sound, they still require validation in practice; we expect the need for higher accuracy, especially for predictions requiring a longer prediction horizon. Finally, the limitation of this work is that we only demonstrated the feasibility on MHS. Although the approach itself is generic and can be applied to event logs from various domains besides MHS, it does not take into account intra-case features of individual cases that are crucial for PPM of business processes. For adopting our approach for business processes, we aim to combine features of both aggregate PS-based and case-based PPM in future work.

## Acknowledgment

The research leading to these results has received funding from Vanderlande in the project "Process Mining in Logistics".

## References

[1] T. Ahmed, T. B. Pedersen, T. Calders, and H. Lu, "Online risk prediction for indoor moving objects," in *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, vol. 1, June 2016, pp. 102–111.

[2] J. Skorupski, P. Uchroński, and A. Łach, "A method of hold baggage security screening system throughput analysis with an application for a medium-sized airport," *Transportation Research Part C: Emerging Technologies*, vol. 88, pp. 52 – 73, 2018.

[3] S. Nahavandi, B. Gunn, M. Johnstone, and D. Creighton, "Modelling and simulation of large and complex systems for airport baggage handling security," in *Intelligent Computing*, K. Arai, S. Kapoor, and R. Bhatia, Eds. Cham: Springer International Publishing, 2019, pp. 1055–1067.

[4] A. Khosravi, S. Nahavandi, and D. Creighton, "Estimating performance indexes of a baggage handling system using metamodels," *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 1 – 6, 03 2009.

[5] A. E. Márquez-Chamorro, M. Resinas, and A. Ruiz-Cortés, "Predictive monitoring of business processes: A survey," *IEEE Transactions on Services Computing*, vol. 11, no. 6, pp. 962–977, Nov 2018.

[6] V. Denisov, D. Fahland, and W. M. P. van der Aalst, "Unbiased, fine-grained description of processes performance from event data," in *Business Process Management*, M. Weske, M. Montali, I. Weber, and J. vom Brocke, Eds. Cham: Springer International Publishing, 2018, pp. 139–157.

[7] A.Senderovich, C.D.Francescomarino, and F.M.Maggi, "From knowledge-driven to data-driven inter-case feature encoding in predictive process monitoring," *Information Systems*, 2019. [Online]. Available: https://doi.org/10.1016/j.is.2019.01.007

[8] B. F. van Dongen, R. A. Crooy, and W. M. P. van der Aalst, "Cycle time prediction: When will this case finally be finished?" in *OTM Conferences*, 2008.

[9] W. M. P. van der Aalst, H. Schonenberg, and M. Song, "Time prediction based on process mining," *Inf. Syst.*, vol. 36, pp. 450–475, 2011.

[10] F. Folino, M. Guarascio, and L. Pontieri, "Discovering high-level performance models for ticket resolution processes," in *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, R. Meersman, H. Panetto, T. Dillon, J. Eder, Z. Bellahsene, N. Ritter, P. De Leenheer, and D. Dou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 275–282.

[11] M. Polato, A. Sperduti, A. Burattin, and M. de Leoni, "Time and activity sequence prediction of business process instances," *Computing*, pp. 1–27, 2018.

[12] G. T. Lakshmanan, D. Shamsi, Y. N. Doganata, M. Unuvar, and R. Khalaf, "A markov prediction model for data-driven semi-structured business processes," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 97–126, Jan 2015.

[13] N. Tax, I. Verenich, M. L. Rosa, and M. Dumas, "Predictive business process monitoring with LSTM neural networks," in *CAiSE 2017*, ser. LNCS, vol. 10253. Springer, 2017, pp. 477–492.

[14] A. Rogge-Solti, W. M. van der Aalst, and M. Weske, "Discovering stochastic Petri nets with arbitrary delay distributions from event logs," in *BPM Workshops 2013*, ser. LNBIP, vol. 171. Springer, 2014, pp. 15–27.

[15] A. Senderovich, A. Rogge-Solti, A. Gal, J. Mendling, A. Mandelbaum, S. Kadish, and C. A. Bunnell, "Data-driven performance analysis of scheduled processes," in *BPM 2015*, ser. LNCS, vol. 9253. Springer, 2015, pp. 35–52.

[16] A. Rogge-Solti and M. Weske, "Prediction of business process durations using non-markovian stochastic petri nets," *Inf. Syst.*, vol. 54, pp. 1–14, 2015.

[17] A. Senderovich, M. Weidlich, A. Gal, and A. Mandelbaum, "Queue mining for delay prediction in multi-class service processes," *Inf. Syst.*, vol. 53, pp. 278–295, 2015.

[18] A. Cuzzocrea, F. Folino, M. Guarascio, and L. Pontieri, "Predictive monitoring of temporally-aggregated performance indicators of business processes against low-level streaming events," *Information Systems*, vol. 81, pp. 236 – 266, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306437917301023

[19] C. Malandri, M. Briccoli, L. Mantecchini, and F. Paganelli, "A discrete event simulation model for inbound baggage handling," *Transportation Research Procedia*, vol. 35, pp. 295 – 304, 2018, iNAIR 2018.

[20] T. G. Dietterich, "Machine learning for sequential data: A review," in *Structural, Syntactic, and Statistical Pattern Recognition*, T. Caelli, A. Amin, R. P. W. Duin, D. de Ridder, and M. Kamel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 15–30.