BACHELOR

A Process-Aware Perspective on the Use of the Performance Spectrum in Predictive Process Monitoring of Business Processes

Costache, Ioana

Award date:
2021

Department of Mathematics and Computer Science
Process Analytics

A Process-Aware Perspective on the Use of the Performance Spectrum in Predictive Process Monitoring of Business Processes

Bachelor End Project Report

Ioana Costache

Supervisors:
Dirk Fahland

9-July-2021

Abstract

The prediction of aggregate process performance indicators (PPIs) is a highly relevant task in business processes where cases are interdependent (e.g. when they are competing for shared resources). Poor monitoring of business processes can lead to unfavorable business outcomes, such as low employee retention and morale, decrease in revenue and even business failure.

Previous research investigates and introduces methodologies for inter-case feature encoding for the prediction of PPIs. The so-called DDE methodology has proven to be suitable for the prediction of case-specific PPIs in business processes. The so-called PS-based methodology has been tested for the prediction of aggregate PPIs in automated Material Handling System (MHS) processes. This paper investigates the adequacy of the PS-based inter-case feature encoding on business processes and proposes an extension to the approach: the Process-aware PS-based PPM methodology.

The extended approach proposes the use of process models along with the Performance Spectrum for the task of inter-case feature selection for prediction of aggregate PPIs of business processes. The outcome of the prediction task indicates that the extended methodology is not sufficient for the prediction of highly volatile aggregate metrics, supporting further research.

Contents

Chapter 1

Introduction

1.1 Context and Topic

The prediction of aggregated performance metrics is highly relevant in business processes, as mismanagement of such processes can lead to process performance bottlenecks that can ultimately result in business failure. [7] Predictive Process Monitoring (PPM) is a subfield of Process Mining preoccupied with the prediction of Process Performance Indicators (PPIs) for running cases. It does so by leveraging historic process performance data to predict and continuously monitor the performance of a business process [8].

State of the art Predictive Process Monitoring techniques for business processes often assume process performance stationarity and case independence [8]. However, these assumptions have been proven to be violated in practice. In fact, most definitions mention interdependence of cases as a key characteristic of business processes [20]. Additionally, the assumption of performance stationary has been refuted with the help of the recent development of the Performance Spectrum [5]. Prediction of Process Performance Indicators (PPI) can be challenging when using traditional techniques which assume case independence when cases are interdependent (e.g. when they are competing for resources).

Recent advancements in the area of PPM have introduced inter-case feature encoding techniques for performance prediction tasks. Existing approaches use inter-case features for the prediction of the remaining time until case completion of individual cases [12] or for the prediction of aggregated PPIs in highly automated Material Handling System (MHS) processes. Current techniques have not been tested on the prediction of aggregated PPIs in non-automated business processes (such as workload).

1.2 State of the Art

Business processes are key business components that occur at all organizational levels within a company. They incorporate systems, data, and resources within and across organizations, and mismanagement of business processes can lead to production bottlenecks and inconsistent workload, which in turn, can cause decreased productivity and business output, decreased revenue, low employee morale and ultimately business failure. [7]

The current state of the art inter-case feature encoding technique for business processes is the data-driven (DDE) approach [12]. The approach uses similarity-based inter-case features to predict the completion time of singular cases. Nevertheless, in reality, aggregate PPIs, such as expected workload at a given moment in time, are highly relevant for business processes [8], but the DDE approach has not been applied on the prediction of aggregate PPIs.

An approach to inter-case feature encoding for aggregate PPI prediction has been developed by Denisov, Fahland and van der Aalst [4]. Their methodology uses the Performance Spectrum for inter-case feature selection for process performance prediction in inter-dependent cases.

The Performance Spectrum is a data structure which allows the visualization of time performance of all process steps across time ([5]). Hence, the Performance Spectrum allows the visualization of start and end time (and consequently, the pending interval) of each process segment (pair of activities), capturing interdepenendencies between cases. It allows for a thorough understanding of aggregate performance indicators of non-stationary processes and of processes in which cases are not independent (for example, when cases are competing for resources). While the PS-based methodology introduced by Denisov, Fahland and can der Aalst has been tested on Material Handling Systems (MHS), the applicability of the approach on non-automated business processes has not been investigated.

This research aims to assess the suitability of the already existing PS-based PPM methodology for aggregate PPIs prediction in business processes. The paper introduces a proposed extension of the PS-based PPM methodology which allows for the interaction between business process rules and the PS-based approach of capturing inter-case features. This will ultimately be used as input to try to achieve a more accurate process performance prediction model without trading off business process awareness.

1.3 Research Question

Formally, the central research topic of the paper can be summarized by the following research question:

To what extent is the PS-based Predictive Process Monitoring approach suitable for non-automated business processes?

To assess the suitability of the PS-based PPM approach for non-automated business processes, we first introduce two assumptions, which stem from the discussion of the PS-based methodology limitation [4]:

1. Applying the methodology on business processes requires the formal introduction of process models into the methodology.
2. Applying the methodology on business processes requires an additional step concerned with the selection and extraction intra-case feature selection.

For the purpose of answering the central question, the following subquestions are introduced:

1. Can each step in the PS-based PPM methodology be applied to business processes?
2. Can the order of steps in the PS-based PPM methodology be maintained when applied on business processes?
3. Are the PS-based PPM methodology steps sufficient when applied to business processes?
    - Does the methodology require the introduction of formal process modelling? If so, at which stage of the methodology is the addition necessary?
    - Does the methodology require the introduction of intra-case feature selection? If so, at which stage of the methodology is the addition necessary?
4. Does the (updated) PS-based PPM methodology produce sufficiently accurate prediction results?

In broader terms, the first three subquestions concern the capability of the methodology of being executed in its original form (i.e. by executing each step successfully, without the necessity of additional steps, while maintaining the same order of steps). For further reference, this property of the methodology will be referred to as the "operability" property. On the other hand, the last sub-question regards the suitability of the (updated) method after the required modifications identified had been applied. Suitability is the ability of the methodology to yield satisfactory prediction results.

1.4 Approach

The approach followed to achieve the goal of the paper follows the same structure as the approach introduced by Denisov, Fahland and van der Aalst. Given that the goal of this research is to test the adequacy of the already existing method, the PS-based approach has been followed step-by-step (i.e. by following the exact same sequence of steps) [4]. The encountering of roadblocks (or lack thereof) while following the step sequence is an indicator of original model (in)operability.

For business process understanding and pre-processing purposes, the ProM Framework [16] has been used in the first stages of the approach. Its supported plugins support the visualization of the Performance Spectrum, the task of process modelling and filtering of event logs.

Further, the Python implementation of the Detailed Performance Spectrum [18] has been used for visualization of performance inter-dependencies purposes, as an aid for selection of candidate segments. The Aggregate PS of the target and historic segments of interest has been computed directly from the event log.

The final stage of the methodology concerns the modelling task. The Machine Learning problem is the prediction of an identified aggregate PPI of interest (such as workload). The Aggregated PS has allowed for the translation of the identified prediction task of interest into a multivariate time series forecasting problem. While assessing the model performance, we will focus both on overall workload prediction and peak workload detection. Separate assessment of peak workload detection is relevant because sufficient prediction of extreme loads of work is crucial in proper preventive management of business operations.

The outcome of both tasks will be, in turn, used as an indicator of the adequacy of the (updated) PS-based PPM approach as a whole.

Evaluation of the entire methodology requires evaluation of two characteristics: operability on business processes and suitability for business processes. Questions 1-3, as outlined in Section 1.3 assess the operability and will be tested by following the already existing PS-based PPM approach step-by-step. If the exact sequence of the approach steps can be applied to the selected business process without further modifications (i.e. all steps can be followed in the determined order and no additional steps are required), the methodology is regarded as operable. If not, it will be adapted to be operable on business processes.

Once the methodology has been adapted to be operable on business processes, the suitability will be assessed by the evaluation of the prediction results. The suitability evaluation task requires training and evaluating several Machine Learning algorithms concerned with workload prediction. For this purpose, performance metrics such as Root Mean Squared Error (RMSE- for overall prediction evaluation) and Recall (for peak detection evaluation) will be used. The RMSE has been chosen as the error metric of choice because it gives more weight to larger residuals, which makes it suitable for prediction tasks where large residuals are not ideal (such as the case of workload prediction). These error metrics are the evaluation starting points. Starting from the general error metrics, for a more detailed evaluation, we will make use of several plots to conduct qualitative analysis to pinpoint specific situations in which the methodology underperforms.

1.5 Findings

Results

After applying the methodology, the results suggest the addition of 3 mandatory steps before the selection of the target segments (the first step of the proposed methodology) and 1 optional step after the extraction of inter-case features. The 2 mandatory steps are: business understanding and pre-processing/ filtering of event log. In summary, by following the PS-based methodology step by step, the results show that in the context of business processes:

1. All existing steps can be maintained.

2. The order of the steps can be maintained.

3. Formal introduction of process models is required from the beginning of the methodology.

4. Intra-case feature selection is not required for mere method operability, but could be required for method suitability for business processes.

Applying the method on event log X we have identified the aggregate segment for prediction, with an average value workload value of 10.37 and 48 peaks. Additionally, we have identified the corresponding features and learned the model. To do so, the evaluation stage requires the introduction of a formal definition of a workload peak which will be used for labelling the observed values and the prediction output.

In the context of the workload prediction task, the best performing prediction algorithm yielded prediction with a Root Mean Squared Error (RMSE) of 6.82, compared to the 8.62 of the naive baseline and the average workload value of 10.37.

We can visualized the overall prediction results by plotting the observed test values together with the predicted values of each model. This helps with assessing prediction patterns and identifying in what kind of scenarios the model underperforms. Additionally, to identify which features might cause the model to underperform, we can plot features of the Historic Spectrum next to the response variable of the Target Spectrum. This allows for assessing the correlation between the predictors and the response variable by visualizing common trends.

For the prediction of peaks, model performance can be visualized with the help of Confusion Matrices. The peak detection recall of the best performing model (3-layer 1D CNN) is 41.37% for a 2-class labelling.

While the methodology performs better than the baseline, indicated some added value from the PS-based features, it is not sufficient in the context of the analysis.

Interpretation

Following the results of the analysis, the methodology is accounted as not operable on business processes. It has been determined that the use of process models is required throughout the entire course of a PPM project in a business setting. Moreover, the analysis results show that process models are necessary from the very first steps of a proper methodology for business processes, as they are essential for process understanding. Thus, due to the absence of this formal step in the methodology, the original PS-based methodology for feature extraction and selection in its original form cannot be applied on business processes.

The proposed updated operable methodology has been in turn proven to be inadequate for the prediction of aggregate PPIs in business processes. In the context of workload prediction, the methodology has been proven to perform poorly. While the best predictors perform generally better than the naive baseline which only accounts for past values of the outcome variable in the Target Spectrum, peak day prediction in processes with high workload volatility over time is unsatisfactory. In business processes, proper prediction of extreme values (such as peaks in workload) is more critical than the overall predictive outcome, because the adequate detection of peaks can diminish the risk of process performance bottlenecks. Given the unsatisfactory result of the peak detection task, it can be concluded the the updated methodology is not suitable for business processes.

Chapter 2

Background

2.1 Preliminaries

Definitions

We first formally introduce definitions, mathematical equations as formerly defined by Denisov, Fahland and van der Aalst [5].

Process Segment (a,b): A process segment $(a, b) \in A \times A$ (where A is the set of activities in the event log) describes a pair of activities a and b. An observation in a process segment is a specific case going through the segment (a step from activity a to activity b of a particular trace). In the Detailed Performance Spectrum, observations in a process segment are visualized with regards to their duration (see Figure 2.1)

Performance Classifier: A performance classifier maps the two events making up a segment (a,b) into a performance class with respect to a chosen performance metric and the performance of all the other segments in the event log. The classification is determined by the mapping function:

$$\mathbb{C} : \mathcal{E} \times \mathcal{E} \times 2^{\mathcal{L}} \rightarrow C \tag{2.1}$$

, where $\mathcal{E}$ is the universe of events, $\mathcal{L}$ is the universe of cases, and C is the set of performance classes. An example of a performance classification function is $\mathbb{C}(t_b, t_a, L)$ which maps events to a class $c \in C$ with respect to the duration between events a and b (duration of a observation in a segment) relative to the duration of all segments in the event log. The determination of performance classes depends on the needs of the user. A common set of classes used for classification is based on the quartiles of the set of segment observations durations.

Detailed Performance Spectrum: The Detailed Performance Spectrum is a data structure which visualizes variability of durations in a segment across cases and time. It visualizes segment observations from $A \times T$, where A is the set of activities in the event log and T is the set of timestamps of each event occurrence.

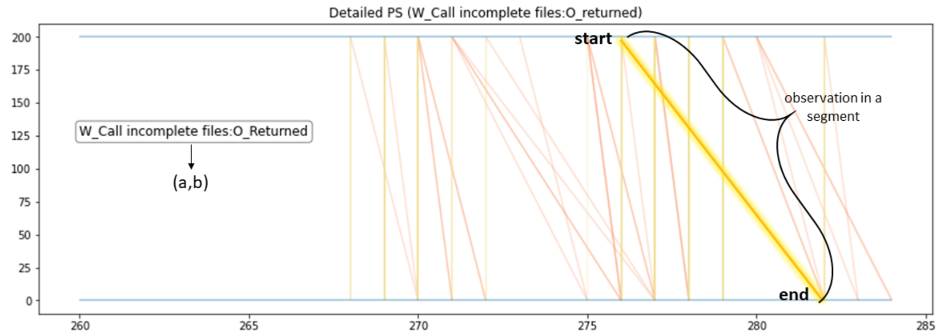


Figure 2.1: Example Detailed PS corresponding to the event log in Appendix A

Bin (b_j): A bin $b_j \in b_1, .., b_m$ (where m is the number of total bins) of set size p (bin size) is the particular record of the time unit (corresponding to the chosen bin size) used for aggregation when computing the Aggregate Performance Spectrum. A bin corresponds most often to a standard time unit such as a particular day, minute, hour etc.

PS channel (ch): The PS channel, by formal definition $ch = (\mathbb{C}, g, p)$ encompasses the parameters describing the bins: $\mathbb{C}$ representing the performance classification, g denoting the grouping (start, pending or end) and p, which stands for the time period. In turn, the PS channel is a parameter of the aggregation vector $S_L((a, b), ch, j)$ which is the building block of the Aggregated Performance Spectrum, where j is the bin number and (a, b) represents the formerly defined process segment.

Aggregation vector (v_j**)**: The aggregation vector counts how often each performance class $c^i \in \mathbb{C}$ has occurred in bin b_j (i.e. the j^{th} bin) and grouping g, where $g \in \{start, pending, end\}$ and indicates whether the segments have been aggregated on their start time, pending time or end time:

$$v_j = S_L((a, b), \mathbb{C}, g, p, b_j) \tag{2.2}$$

Aggregated Performance Spectrum: The Aggregated Performance Spectrum permits the visualization of aggregate process performance metrics. The aggregate PS of one segment (a,b) can be visualized as a series of stacked barcharts where the bars correspond to the bins of width p and the stacks correspond to the classes. The Aggregated multi-channel PS defines a set of segments $S = \{(a_1, b_1), (a_2, b_2), ...\}$ and a set of channels $ch_1, .., ch_k$. The Aggregated PS hold performance information in 3 main dimensions:

- the segment (a_i, b_i),
- the channel $ch_l = (C_l, g_l, b_l)$
- the bin b_j at vector $v_j = S_L((a, b), \mathbb{C}, g, p, b_j)$

For the purpose of the prediction task introduced in this paper, a single-channel Aggregated PS has been defined.

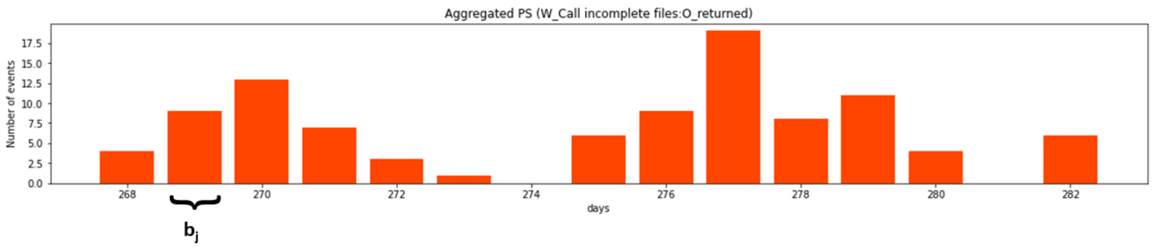


Figure 2.2: Example Aggregated PS corresponding to the event log in Appendix A

Further, we introduce other relevant terms encountered throughout the paper:

PPI: A PPI of Process Performance Indicator is a process-specific KPI. A PPI represents a measurable target that helps an organization measure effectiveness of their processes in achieving operational goals.

The PS-based PPM methodology

For the purpose of this research question, we repeat verbatim the PS-based PPM methodology as formulated by Denisov, Fahland and van der Aalst [4]:

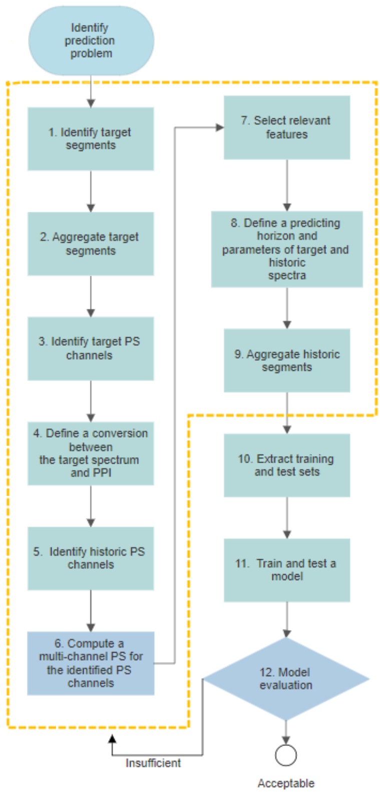
Figure 2.3: The configuration of the historical and target spectra, extracted from [4]

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 $(a_1, b_1), ..., (a_n, b_n)$ can be aggregated into a new aggregated segment (a^*, b^*) by relabeling $a_i \to a^*, b^*$ 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. 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 another grouping pending with a duration-based classifier is required. Then in **Step 4**, given the identified Target Spectrum parameters, a concrete function g 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 s1 causes a higher load on target segment st, 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? What is the time interval of the Historic Spectrum that should be used to estimate the Target Spectrum?

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. 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 [6], 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 nontarget PS channels, aggregation rules, selected features, PH and model configuration in order to improve the model accuracy.



“

Figure 2.4: PS-based PPM methodology

2.2 Related Work

As summarized in 1.2, prior to the PS-based inter-case feature encoding technique, the so-called DDE approach has been tested on business processes [12]. This encoding technique assumes no prior knowledge about the process. The approach introduces the use of case proximity metrics to automatically identify similar cases. Their defined proximity metrics concern both aspects related to the control-flow (i.e. the sequence of steps followed by an individual case) and aspects related to temporal distances between the cases. The proposed DDE inter-case feature encoding technique has only been proven to be adequate for the prediction of case-specific PPIs (such as time until case completion), as opposed to aggregate PPIs. [4]

Related work on Process Mining Project Methodologies ([17], [3]), propose the adaptation of common Data Mining Project frameworks (such as CRISP-DM [13] and SEMMA [2]) to meet the needs of process mining-related task. It has been established that commonly used data mining project frameworks have limited support for Process Mining projects, as the field of Process Mining encompasses elements and techniques of both Machine Learning and Data Mining, as well as those found in the Business Process Management discipline [14]. One of the current standard Process Mining project methodology for business processes (PM^2) [17] include the formal steps of process discovery, conformance checking and enhancement after the extraction of the Event Log and prior to the carrying out of Machine Learning/Data Mining task. We have used part of the methodology for process model analysis for the purpose of feature selection, but the methodology is not appropriate because it does not include explicit steps suitable for prediction problems.

Chapter 3

PS-based PPM of credit application processes

The problem investigated by this paper is the problem of assessing the operability and suitability of the PS-based PPM method on business processes, as outlined in Section 1.3. For the purpose of investigating this broad research task, two sub-problems have been identified

1. The process-aware PS-based PPM feature selection methodology, which will be described in subsection 3.1

2. Workload prediction which will be described in subsection 3.2

3.1 Process-aware PS-based feature selection methodology

Before introducing the methodology for feature selection for PPM of aggregate PPIs in business processes, we shall identify a business goal. In broad terms, the determined business goal corresponding to the process is the prediction of workload, particularly the peak values, which could be indicators of work overload. This is related to efficiency of business operations.

To achieve the determined goal, the Inter-Case Performance Prediction Problem [4] will be solved in accordance to the PS-based methodology as outlined by Denisov, Fahland and van der Aalst. Before doing so, we shall recall the assumptions outlined in Section 1.3. The assumptions state the following: the introduction of formal process models and the selection and extraction of intra-case features of individual cases are necessary requirements for an operable methodology in the context of business processes.

As outlined in Section 1.4, for the purpose of assessing the method operability, the steps will be applied as they are presented in the original approach. Obstructions in the application on business processes signal possible inaptness of the approach and we will correct it by updating the methodology to allow for progress through the pipeline.

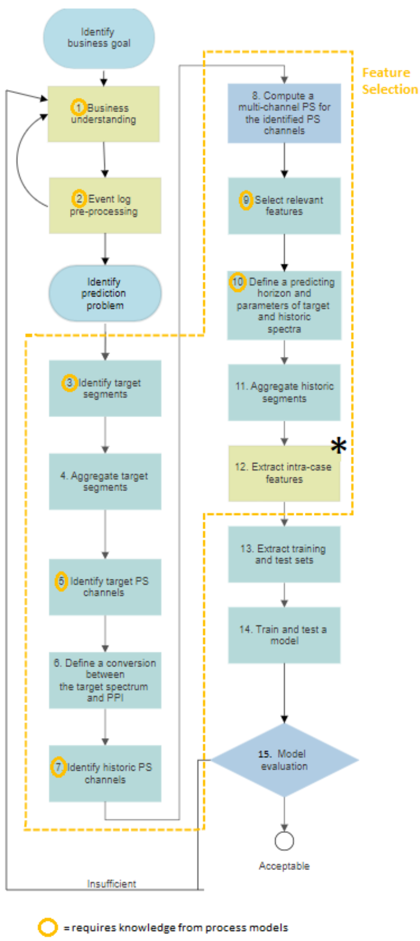


Figure 3.1: Updated methodology for business processes

3.1.1 Data and Business (Process) Understanding

We introduce the **Step 1** of the Process-Aware PS-based PPM methodology: business understanding. The selected business process is the BPI 2017 dataset [15], with information about a credit application process. The original dataset contains 31 509 cases and 1 202 267 event records, varying in frequency. The logged events span over approximately 13 months, from January 4th, 2016 to February 2nd, 2017. The events are split in 3 categories, identifiable by prefix:

- A: application state changes
- O: offer state changes
- W: workflow events

In order to define the specific prediction problem, understanding of the credit application process from a panoramic level is required. Povalyaeva, Khamitov and Fomenko [10] have determined seven main stages of the process, as seen in 3.2:

1. Every trace starts with A_Create_Offer, when a new loan application is filed.
2. The application undergoes routine checks before an offer is made.
3. One or more offers are created by the bank and being communicated to the client.
4. The client sends the documents required by their preferred offer.
5. If the client has failed to send all the required documents or the required documents or the documents are incorrect, the bank notifies the client (W_Call_Incomplete_Files) , asking for the missing documents.
6. If the documents are complete but unacceptable, the bank can offer a more suitable credit option.
7. Client may send documents or cancel the application. This step and the previous one constitute an iterative subprocess. Once the client has sent all the correct documents, the bank makes a decision.

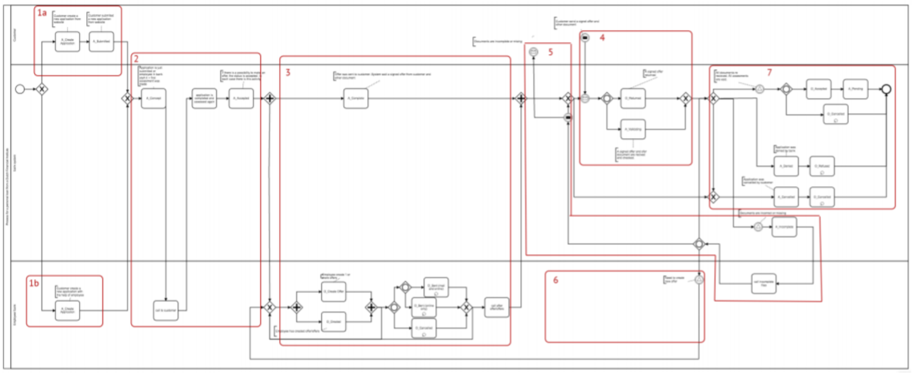


Figure 3.2: BPMN model of the entire process, corresponding to the stages identified by Povalyaeva et al.

Additionally, we have used the Performance Spectrum Miner plugin of the ProM Framework [16] on the complete, unfiltered BPI2017 dataset to identify segments that exhibit highly volatile workload trends. We are interested in volatile trends because such trends signal unpredictable process behaviour which, if not predicted with sufficient accuracy, can result in process and resource mismanagement and, consequently, work overload. Based on the process structure and Aggregate PS , the identified segments of interests are W_Call_Incomplete_Files:O_Create_Offer (seg_1) and W_Call_Incomplete_Files: O_Returned (seg_2). As seen in Figure 3.3, their aggregated performance spectra display volatility and day-of-the-week seasonality, which make them suitable for the prediction task.

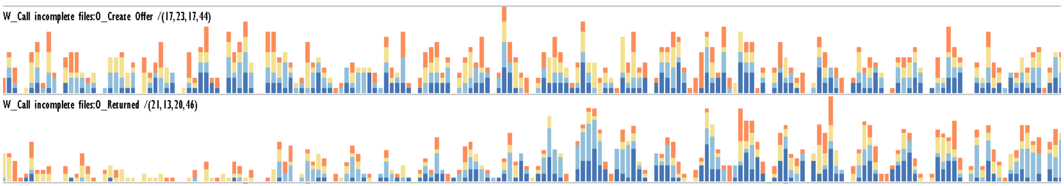


Figure 3.3: Workload volatility in selected segments seg_1 and seg_2

One particularity of business processes is the nature of the the activities with regard to the spread of the workload across the week. As shown in Figure 3.4, the workload amount displays a day-of-the-week seasonal component, corresponding to the common structure of a workweek (increased business activity from Monday to Friday and decreased activity from Saturday to Sunday).

Initially, the selection of segments is based merely on the observations from the aggregated performance spectra. As seen in Figure 3.3, the aggregated performance spectra are depicting periodically repeating rising and falling workload trends where some rising trends are significantly higher. For the prediction task, we need a thorough understanding of the subsequent occurrence of events, particularly of those leading to the target segments. In the case of complex business processes, process understanding is a prerequisite for PS-based feature selection, which is in turn formalized in [4]. Thus, we have determined that the Performance Spectrum alone is not sufficient for the identification of segments of interest, hence, one or more process models are required in the methodology.

Given the above sketched process (as originally seen in [10]) and the patterns visible in Figure 3.3, we have determined that the chosen segments (seg_1, seg_2) are suitable both from a prediction and process perspective, as the identified segments correspond to stages 5-7 of the process, conforming to the general flow. More process modelling will be carried out throughout the experiment, after pre-processing the event log and as support for determining the historic spectrum. For increased understanding of the subprocess of interest, the event log has been filtered to meet the prediction needs, introducing a new step in the methodology: pre-processing.

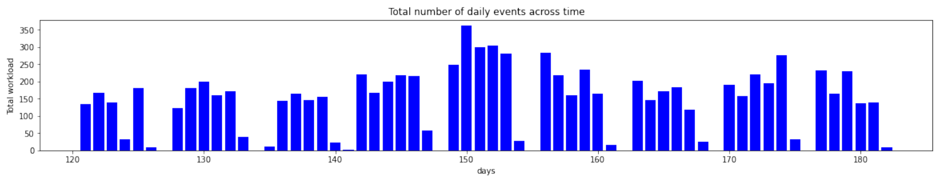


Figure 3.4: Total number of recorded events across 9 weeks

3.1.2 Pre-processing

Step 2 of the updated methodology is Pre-processing. Filtering of the event log has been done using the Filter Event Log plugin of ProM and consists of three main steps:

1. For model simplification purposes, all traces in which the W_Call_ Incomplete_Files event is never followed by O_Create_Offer or O_Returned have been filtered out of the log.

2. Only workflow events classified by their lifecycle transition that occur in at least 80% of the cases have been kept.

(a) All workflow events that were logged in the *scheduled* stage of their lifecycle have been filtered out.

(b) All application and offer state change events have been kept, with the exception of A_Incomplete and O_Created. Such activities have been omitted in order to avoid redundant information, as they almost always directly follow other events that hold very similar information about the state of the application or offer (W_Call_Incomplete_Files and O_Create_Offer, respectively, see Figure 3.5).

3. All subsequent offer and application state changes of same concept have been merged, with the exception of O_Create_Offer, which can give information on the number of different offers that have been forwarded to the client.

   (a) Subsequent workflow events of same concept have not been merged. Instead, the information about the number of consecutive same-concept workflow events has been kept and can be captured as intra-case feature.

After the preprocessing step, the log has been reduced to 2995 traces of 132109 unique events.

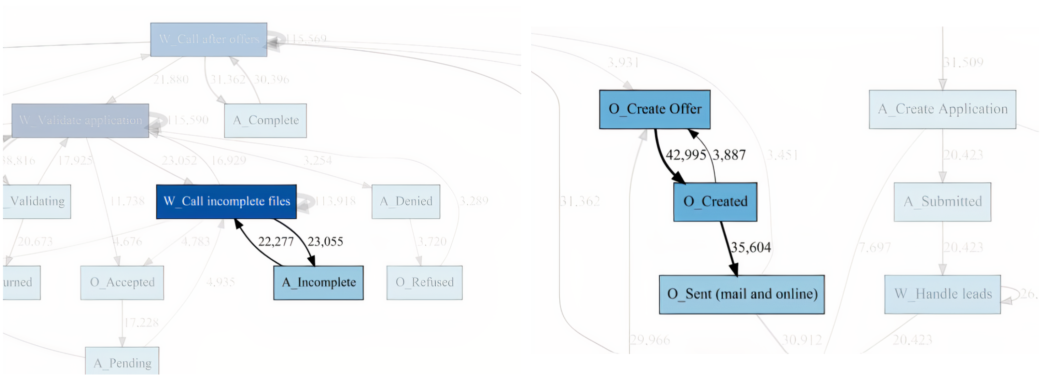


Figure 3.5: Directly-follow graphs depicting redundant information, as defined in sub-step 2.(b) of the pre-processing step. State changes *A_Incomplete* and *O_Created* always succeed *W_Call_incomplete_files* and *O_Create_Offer* respectively, events which hold very similar information about the state of the application or offer.

For improved understanding of event dependencies and control flow, we have discovered and inspected two formal process models. All process models have been discovered with the help of the Interactive data-aware heuristics miner ProM plugin. First, a C-net of the event log with frequency threshold of 40% has been visualized for the representation of the causal dependencies between events. Secondly, while a Causal Net adds valuable information about the control flow of the entire process, additional information about direct event dependencies has been obtained by modelling a Directly-Follows graph of the subprocesses of interest.

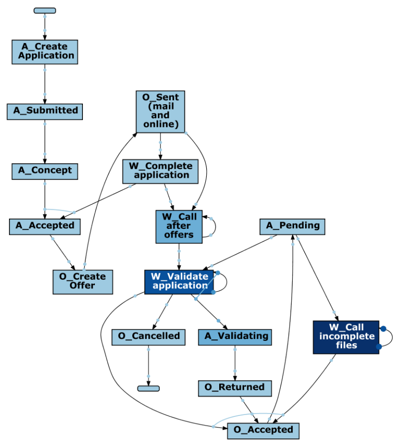


Figure 3.6: C-net used to visualize causal dependencies between cases

Figure 3.6 shows the Causal Net of the entire process, which we have used to identify the possible behaviours of the process. The model shows standard process behaviour in the beginning of the process, corresponding to the automatic checks and acceptance of the application. The events preceding and including O_Create_offer have 100% occurrence probability, meaning that all clients receive at least an offer [10]. An application is completed after the suitable offers have been sent. The application completion is followed by a series of document request and client calls ($W\_Call\_after\_offers$, $W\_Validate\_application$, $W\_Call\_incomplete\_files$) which lead to a series of changes in the state of the application and offer(s). As opposed to the automatic application checks, the documentation check sub-process which takes place after the offer(s) have been forwarded is highly iterative and follows a less definite structure. Thus, while the Causal Net is valuable for understanding the control flow of the entire process from a higher level, we will be zooming into and analyzing the sub-processes of choice in more detail in the next steps.

3.1.3 Inter-case feature selection

For a more systematic assessment of the feature selection method design, the inter-case feature engineering problem has been divided into two subproblems: specification of the Target Spectrum and discovery and computation of the Historic Spectrum.

Target Spectrum

Following the process discovery stage, it has been determined that the two previously identified target segments $W\_Call\_Incomplete\_Files : O\_Create\_Offer$ (seg_1) and $W\_Call\_Incomplete\_Files : O\_Returned$ (seg_2) are feasible from a process perspective as well as from a prediction perspective. The identification of the target segments marks **Step 3** of the updated methodology. The aggregation of the target segments (**Step 4**) has been computed by relabelling seg_1 and seg_2 as seg_t. The implementation, limited to the Detailed Performance Spectrum has been used for the brief inspection of the segments observations. While the Detailed PS implementation can be extended for the Aggregate PS, we can compute segment aggregations directly from the event log, as each unique event has a recorded timestamp.

For **Step 5**, we identify the PS channel. Given that the PPI of interest is the amount of workload at the start of the aggregated segment (i.e. aggregated on activity $W\_Call\_Incomplete\_Files$), the defined PS channel with a single performance class is $ch_{t,w} = \{\mathbb{C}, start, 1\}$, where the performance classifier $\mathbb{C}$ returns the same value for all segment occurrences. Specifically, a bin size of 1 day has been selected due to the nature of the business process, in line with the motivation presented in Section 3.1.1. By doing so, information about the day is being captured in the inter-case feature selection task. Capturing day-specific information in a performance prediction problem is crucial in resource and operational planning, especially when dealing with case interdependencies and resource scarcity [19]. For workload prediction, performance classification in the Target Spectrum does not ultimately matter, thus the defined channel consists of a unique performance class. For **Step 6**, the following PPI conversion can be made:

$$PPI_w = S_w(SEG_{t,w}, ch_{t,w}, T+1) \tag{3.1}$$

, where $SEG_{t,w} = \{seg_1, seg_2\}$, T is the present moment, and w denotes the workload as a PPI.

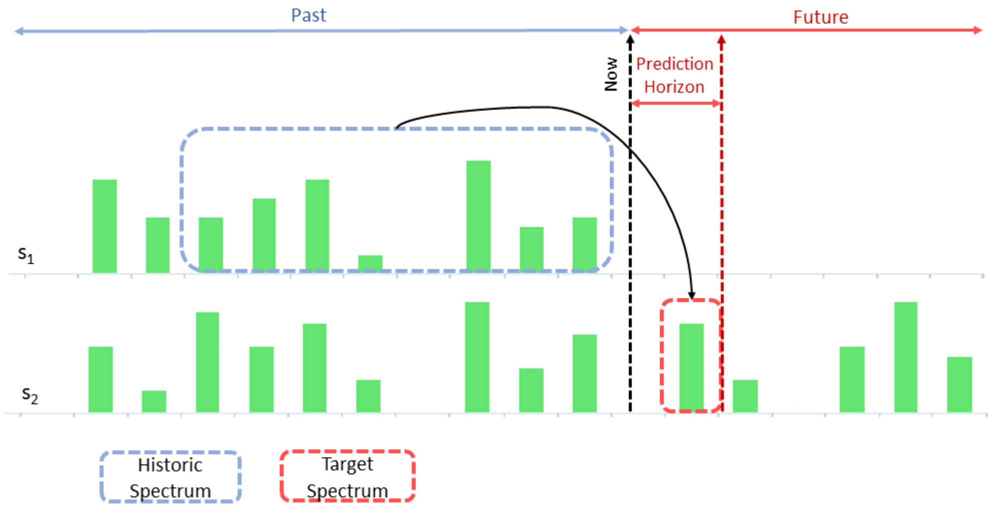


Figure 3.7: Visual representation of the use of the Historic Spectrum for prediction of the Target Spectrum

Historic Spectrum

After identifying the Target Spectrum, we need to define the Historic Spectrum. **Step 7** of the methodology regards the formal definition of the historic PS channel(s). Given the nature of the business process, before defining the historic PS channels, it is essential to properly identify (candidate) historic segments which conform to the execution ordering of the process events. For the identified prediction task, it is suitable and convenient to select segments that capture the activities setting the W_Validate_Application event off. In tune with the process sketched in Section 3.1.1, the W_Validate_Application corresponds to stage 3 of the process described in subsection 3.1.1. The identified candidate segments are:

- W_Call after offers:W_Validate_Application (seg_3)
- O_Returned:W_Validate_Application (seg_4)
- A_Pending:W_Validate_Application (seg_5)
- W_Call_Incomplete_Files:W_Validate_Application (seg_6)
- A_Validating:W_Validate_Application (seg_7)

Following the initial selection of segments, the identified channel, $ch_h = \{\mathbb{C}, end, 1\}$ takes the same bin size as the Target Spectrum channel ch_t, following the same motivation regarding the working day. Similarly to the target channel, the historic channel comprises of a single class. As opposed to the Target Spectrum, segments are aggregated on end time. The segments starting with the identified event have not been chosen for aggregation either because of loops in the process model or because the event on the back end of the segment (in the case of W_Validate_Application:O_Cancelled) represents the sink place (end state) of the process, as seen in Figure 3.8

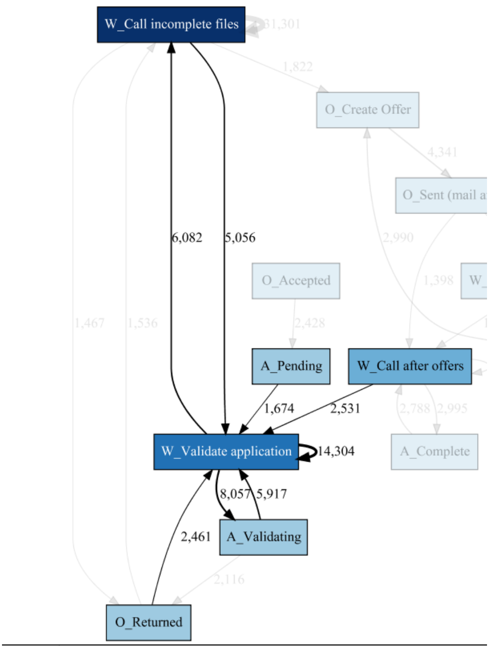


Figure 3.8: Directly-follows model used for candidate historic segments selection

Step 8 requires computing a multi-channel Historic Spectrum. Given the candidate segments and unique identified channel, we can skip this particular step. To improve robustness of the prediction model, we can compute multiple channels and add them to the multi-channel Historic Spectrum. In **Step 9** we identify relevant inter-case features. Besides the cumulative daily workload aggregated on the the end of the identified segments (i.e. number of calls towards clients after applications had been sent), the bins already contain information about the day of the week, week and month of the year. Other inter-case attributes we have derived and used as input features in the Machine Learning algorithms are maximal waiting time between the end state of determined segments, time interval between the first and last segment ending times occurring in a day, which can be an indicator of length of working day, and the total number of unique resources (i.e. employees) allocated to the event corresponding to the end of the process segment in a particular day. **Step 10** is determining the Predicting Horizon. With the endpoint of the Historic Spectrum representing the day before the present moment ($e_h = T - 1$), the Target Spectrum starts and ends one time step in the future ($s_t = T + 1$). Hence, the Predicting Horizon ends one day into the future (i.e. prediction model predicts the target workload values one day into the future, see Figure 3.7). Related to the predicting horizon is the choice of sliding window size. As previously mentioned, we have chosen a window of size 8. The reason is related to the performance of the naive baseline for different window sizes, as seen in Figure 3.9. To ensure a reliable comparative analysis, we have decided to keep the window size constant for all models. A window size of 8 days is suitable, as it captures sufficient overall volatility and day-of-the-week seasonality information without the need to look for trends farther in the past. After identifying the candidate historic segments and selecting relevant features, we aggregate the segments for **Step 11**.

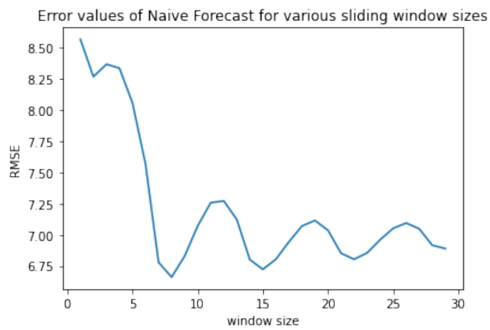


Figure 3.9: Naive baseline error for different window sizes

Inter-case input features	Description
Historic Workload	Total number of events in a bin of the Historic Spectrum
Day of the week	Monday, Tuesday, ..., Sunday
Number of resources	Total number of distinct human resources allocated in a day (bin)
Maximum waiting time	Maximum waiting time between two consecutive events in a bin
Length of work day	Time between the first event and last event of the day

Table 3.1: PS-based Inter-case features

Finally, the following regression function is defined:

$$S_w(SEG_{t,w}, ch_{t,w}, T+1) = f(S_w(SEG_{h,w}, ch_{h,w}, [T-8, T-1])) + R, \quad (3.2)$$

, where $SEG_{h,w}$ represents the set of segments $\{seg_3, ..., seg_7\}$

3.1.4 Intra-case feature selection*

Business process event logs contain valuable information about individual cases which are of relevance for PPM in non-automated processes. The added value of exploiting inter-case dependencies for PPM of case-specific PPIs (i.e. remaining time until completion of individual cases) has already been demonstrated in [12]. In contrasting view, information about individual cases can be of great value for the prediction of aggregate performance metrics is business processes. While the task of intra-case feature encoding and extraction for prediction of aggregate PPIs has not been undertaken in the context of the particular prediction task at hand, this paper introduces a proposal on how to include case-specific information in further research. This makes up for the optional **Step 12** of the methodology.

Consider the following scenario of possible relevance to the specific problem at hand: prediction of the number of daily document request calls. Essentially, requests for additional documents are checks and verification events, which are common in the case of clients with low credit scores who encounter more difficulties when applying for a loan. Could we expect a steep increase in the number of such calls after a period characterised by a high number of applications from clients with bad credit?

Since we have identified the possible relevance of intra-case features for the prediction of aggregate performance metric, the first step that should be taken in order to make the prediction of the aggregate PPI possible is feature encoding. Scientific literature has introduced several intra-case encoding techniques for PPM since the emergence of the discipline. One such techinque is the STEP technique, introduced in [11]. This technique follows similar concepts as common Machine Learning techniques for predicting labels is sequential data [6].

To justify its appropriateness in relation to the PS-based PPM methodology, we introduce two basic concepts of STEP:

1. **Case types.** This concerns the classification of cases based on the intra-case metric of choice. For the distinguish of cases types based on their credit score, we can do one of the following:

    - Manual labelling of cases based on their credit score. The categorization of credit scores is determined by the financial institution offering the loan, which usually follows industry standards.[1]
    - Labelling of cases based on their credit score relative to the credit score values of all other cases in the event log.

2. **Derivation function.** This is essentially a mapping function which maps the classes corresponding to case types. In the context of the Performance Spectrum, the derivation function is equivalent to classifier $\mathbb{C}$, as introduced in Section 2. The only difference is that the categorization is not performance-related and is not necessarily relative to the value of all cases. The mapping function takes the following form

$$f : \mathcal{E}_C \times \mathbb{T} \rightarrow X \tag{3.3}$$

    , where $\mathcal{E}_C$ is the set of events categorized according to their case class types, $\mathbb{T}$ is the set of timestamps recorded in the event log and X is the feature space of interest. In the credit score example, f outputs the number of cases of type $c_i \in \{c_1, ..., c_n\} \in C$ (where n is the number of class types) at their "complete application" stage (i.e. after application is complete and offers have been sent to the client) at time $t \in \mathbb{T}$.

3.2 Workload Prediction

Once we have followed the first stages of the method and updated them to fit the particular feature selection and engineering needs of a business process, we can continue with the modelling task.

3.2.1 Modelling

Extract training and test set

With the help of the Performance Spectrum, the Inter-Case Performance Prediction Problem can be translated to a multivariate time series forecasting problem. As previously shown, a window size of 8 has been used for the prediction purpose. In other words, the observations at times [T-8, T-1] of the Historic Spectrum will be used to predict the workload a time T+1 of the Target Spectrum, where T is the present moment (now).

As a first step, a temporal train-test split of approximately 70:30 has been made, in which the first 70% observations are part of the training set and the last 30% part of the test set. The choice for the split ratio stems from the relatively low number of observations (396 days). A 70:30 split is sufficient for capturing volatility and the day-of-the week seasonal component in both sets. Although activity in many businesses presents yearly seasonality (for instance, decreased business activity in the first and last weeks of the year and increased business activity in the beginning of the 4th Quarter), the dataset does not capture yearly trends, as only 13 months have been recorded. Extracting training and test sets represents **Step 13** of the updated methodology.

Training and testing the model

In **Step 14** we train and test the model. To capture information contained in the Historic Spectrum, we have used a sliding window of 8 time units (days) corresponding to the Historic Spectrum for the supervised learning models, as well as for the naive baseline. We have determined the sliding window of 8 time units to be the most appropriate window size for evaluation purposes. In other words, the model is mapping the information available in windows at times $t_i \in [t_{p-8}, t_{p-1}]$, to predict the total workload at time t_{p+1} where t_{p+1} is the time of the prediction. The Machine Learning models used for prediction are a Multiple Linear Regression (MLR) and 3-layer 1D Convolutional Neural Network (1D CNN). We have chosen the MLR because it is a white-box model where the process is transparent and the role of the features is observable. We have chosen the CNN due to its suitability for time series forecasting [9]. We motivate the choice of 1D convolutions because of the one-dimensional shape of the data (the only dimension is the time dimension), as opposed to other more popular uses of CNNs (such as image classification).

Further, we introduce a formal definition of a peak in order to label the observed values and the predicted values. We label an observed or predicted value as a peak if the workload value the previous and the following day are both at least 10% lower than the workload observed or predicted at the current day ($w_{t-1} + 0.1 * w_{t-1} < w_t$ and $w_{t+1} + 0.1 * w_{t+1} < w_t$).

Chapter 4

Evaluation

The approach to investigating the problem introduced in section 1.3 has been laid out in Chapter 3. The purpose of this Chapter is to evaluate the quality of the approach by interpreting the experiment results. The evaluation of the model represents **Step 15** of the updated methodology.

Evaluation of the methodology is based on:

- Evaluation of results of reasearch sub-questions Q1-Q3: reflection on the applicability of steps, completeness and sequential structure of the methodology after having followed it step by step. We call this the feature engineering method evaluation.

- Evaluation of results of research sub-question Q4: Workload prediction results. We call this the prediction model evaluation.

4.1 Objective

The objective of the evaluation is to assess the quality of the methods that have been used throughout the experiment in order to achieve the research objective. As formally explained in 1.3, the aim of the paper is to determine the adequacy of the PS-based PPM methodology in the context of business processes. Following up the adequacy assessment, it will be determined what additions, tweaks and improvements can be done to the already existing methodology in order to fit the needs of performance metric predictions in business processes.

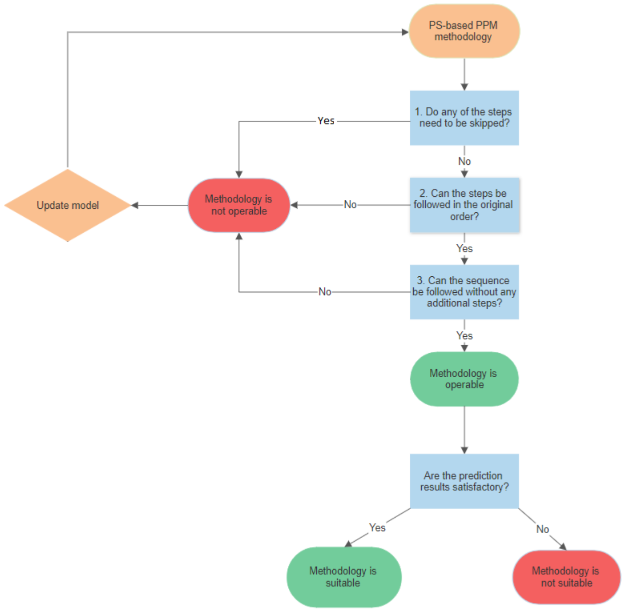


Figure 4.1: Evaluation setup of the PS-based PPM approach

4.2 Feature engineering method evaluation

4.2.1 Setup

The PS-based PPM approach is considered adequate for business processes if it satisfies the two criteria: operability and suitability. To evaluate the operability of the model, the methodology needs to be successfully applied to the business process of interest in its original form. More specifically, the approach has to be applied maintaining the exact step sequence, without skipping any of the steps and without requiring any additional steps.

To evaluate the applicability of the existing approach on the prediction of business aggregate PPIs, the method will be applied exactly as layed out in [5]. If we encounter any obstructions in the process which hinder the progress through the course of the methodology step sequence, we will extend the method with the necessary steps or modifications. The result of the application of the methodology on the BPI2017 dataset is the PS-based PPM methodology extended for aggregate business PPIs.

The Modelling stage (train and test set extraction and training and testing the ML prediction model) will be applied after the methodology had already been adjusted and will be evaluated in more detail in Section 4.3.

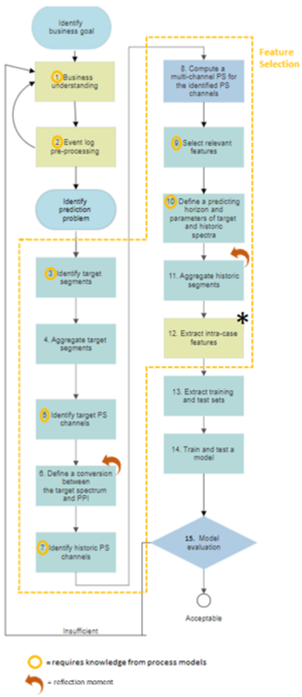


Figure 4.2: Process-aware updated methodology with indication of evaluation moments

4.2.2 Execution

We executed the process-aware PS-based feature selection approach as laid out in Section 3.1. For proper reflection on the operability of the method, we have identified two assessment moments. These assessment moments have taken place right after the identification of the Target Spectrum and the computation of the Historic Spectrum and their purpose is providing a retrospective look into the sequence of the steps required for determining the Historic and Target Spectra, as indicated in Figure 4.2. We have identified the same reflection setup for both of the assessment moments. Thus, each of the first three research questions have been answered twice, once for the computation of the Target Spectrum and once for the computation of the Historic Spectrum.

4.2.3 Results

After following the methodology step by step, we have drawn the following results observations:

1. Identification of the prediction problem cannot be done without the understanding of the process aided by the introduction of process models. As exemplified in Subsection 3.1.1, for this experiment, even before engaging into the pre-processing task, a panoramic BPMN model was required. During the execution of the pre-processing step, we have investigated several subprocesses in more detail with the use of Directly-Follows graphs.

2. Selection of target and historic segments cannot be done without the formal introduction of process models. To understand the direct relationships between activities, as shown in Sections 3.1.2 and 3.1.3, we have used Directly-Follows Graphs.

3. None of the steps in the original methodology have to be skipped when applying them to business processes.

4. The steps in the original methodology can be maintained in the same order.

5. The methodology is executable on business processes without the introduction of intra-case features into the analysis.

4.2.4 Discussion

For this purpose, the first part of the discussion aims to present the shortcomings of the original methodology and the proposed extension. The interpretation is based on the answers to the first 3 research sub-questions and the barriers encountered while following the method.

Based on the observations following the discovery of the process, we can answer the first part of the third subquestion. Feature selection in PPM tasks in business processes requires a process-aware methodology. Therefore, in line with standard Process Mining Project Methodology (PM^2) [17], process discovery is among the preparatory steps in a Process Mining project and should be carried out before determining the target segments. More specifically, as a best practice first step visual analysis, we should use of one or several initial comprehensive process models, followed by the basic pre-processing of the event log. Due to the pivotal importance of process discovery for understanding of fuzzy business processes, this first section of the methodology is highly iterative. It is relevant to note that the use of process models aims to facilitate the selection of target and historic segments by providing visual insights into the control flow of the process, therefore, a panoramic understanding of the entire process is valuable. However, not all subprocesses require detailed understanding in the context of a limited prediction task. For instance, in our case, we are not interested in details regarding all the possible end states of the process. Additionally, the first stages of the process concern routine application checks that largely follow a linear sequence. Details about the automatic application checks are therefore not crucial for the prediction problem.

Given the interpretation above, we have identified 2 additional steps necessary for a method to be operable on business processes:

1. business (process) understanding

2. event log pre-processing and filtering

Both of these steps require extensive use of formal process models.

Further, the presented approach follows the specification of the the Target Spectrum. It can be concluded that the steps of the original methodology corresponding to the Target Spectrum can be applied to business processes maintaining the exact sequence.

With regards to the Historic Spectrum, as demonstrated, we can conduct the discovery of the Historic Spectrum and feature selection for the forecasting problem following the corresponding steps of the original version of the methodology. While the problem instance presented has followed the methodology step by step, depending on the prediction needs, we can make changes to the sequence. Some steps can be skipped altogether (e.g. Step 8), while others can only be followed and implemented correctly by leveraging the insights provided by the process models (e.g. Steps 7, 9, 10).

As a general observation, the inter-case feature selection methodology for business processes requires more flexibility in the order of steps. Additionally, although it has been established that process discovery should precede any other steps, to ensure proper process understanding, we can re-introduce the use of process models at any time during the inter-case feature selection stage (see Figure 3.1)

Given the reflection on the methodology as discussed above, we can answer the first 3 research sub-questions.

1. Firstly, we have determined that all existing steps can be applied.

2. Secondly, the order of the steps can be maintained.

3. Thirdly, additional steps are required. Particularly, the inclusion of process model is necessary from the beginning of the approach, as they play a pivotal role in the extension of the methodology with 2 new steps that are required for the operability of the approach on business processes.

4. However, the inclusion of intra-case features into the analysis is not necessary for the operability of the methodology. The extent to which they add value to the predictive task is to be determined. Based on the conclusive remarks just presented, the original methodology is deemed as not operable on business processes.

Thus, the model needs to be extended with 3 additional steps:

1. Business understanding

2. Event log pre-processing

3. (Optional, not included in this particular experiment) Intra-case feature selection

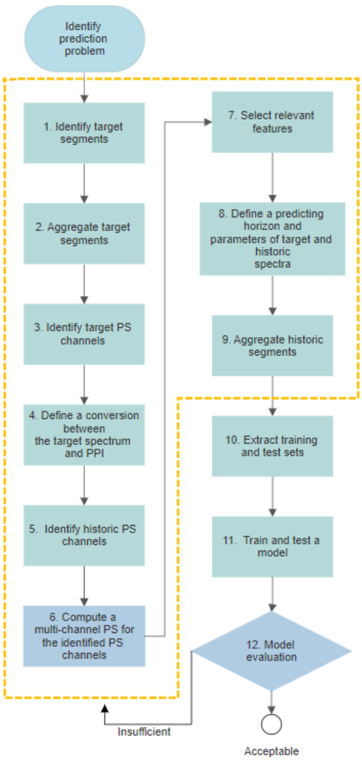


Figure 4.3: PS-based PPM methodology

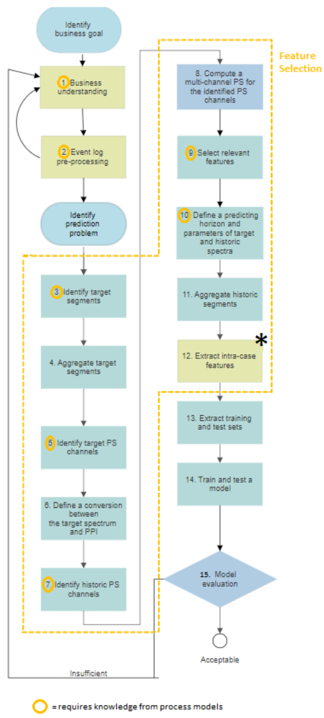


Figure 4.4: Updated methodology for business processes

We can now apply the extended model on business processes (is operable, as opposed to the original one) and we can discuss the results of the modelling phase.

4.3 Prediction model evaluation

4.3.1 Setup

The assessment of the workload prediction task results encompasses two metric perspectives: overall workload prediction and workload peak detection.

The overall workload prediction results can be assessed by means of error metrics and visual analysis. More specifically, we will measure the performance of the Supervised Learning models used for training and testing in terms of RMSE, where the RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(Predicted_i - Observed_i)} \tag{4.1}$$

The RMSE serves as a comparative error measure, as an aid for selecting the best performing model. We will compare the two ML algorithms to a naive baseline, which computes the predicted workload as the average of the workload at times $t_i \in [T-8, T-1]$.

However, a robust performance analysis cannot be limited to an error metric averaged over the entire test set. For a complete analysis, qualitative analysis by visualizing the observed values together with the predicted values is required.

Additionally, in business processes, predictions of extreme values of workload (i.e. the prediction of work overload) is the most important. Sufficient and timely prediction of work overload allows for the development preventive measures to avoid process performance bottlenecks. For analysis of the model performance, we have introduced a formal definition of a peak in order to label the observed values and the predicted values. The evaluation results have been visualized with the help of a Confusion Matrix which has in turn been used for the calculation of the recall.

For peak prediction performance, a recall threshold of 50% has been set prior to the experiment. The peak prediction results are deemed as acceptable only if they exceed the threshold. The reason behind the selection of the threshold value is the number of classes (2- peak and non-peak). Strictly exceeding the 50% recall threshold is the minimal requirement for the detection to be considered acceptable, as it essentially means that the majority of the peaks have been detected. For calculating the recall, the observed values and the predictions have been labelled as peaks if the workload value the previous and the following day are both at least 10% lower than the workload observed or predicted at the current day ($w_{t-1} + 0.1 * w_{t-1} < w_t$ and $w_{t+1} + 0.1 * w_{t+1} < w_t$). Additionally, we only consider peaks the days in which the total number of events exceeds 10 (higher than the average). After labelling, a Confusion Matrix has been computed and given that the analysis interest surrounds the correct prediction of workload peaks, we have calculated the recall.

$$Recall = \frac{\#TruePositives}{\#FalseNegatives + \#TruePositives}. \tag{4.2}$$

At the final stage of the evaluation, if both the overall workload prediction and the workload peak prediction are regarded as satisfactory, the (extended) approach will be regarded as suitable. If any of the model performance metrics is unsatisfactory, the approach is regarded as unsuitable.

Evaluating the performance of the methodology for workload prediction requires comparison with the performance of a naive baseline model, according to which the predicted outcome at time T+1 is the average of the workload at times $t_i \in [T-8, T-1]$. This interval corresponds to the start and end times of the Historic Spectrum in relation to the present moment T. We have maintained the values for sliding window size, prediction horizon size and lag for all models for the purpose of creating a reliable comparison basis.

The chosen Supervised Learning models for prediction were a Multiple Linear Regression (MLR) and a 3-layer 1D Convolutional Neural Network. The reason for choosing these models is due to their applicability to time series forecasting [9]. We have particularly chosen the MLR due the nature of the algorithm. Due to the explainability of the algorithm, the role of the input features in the prediction task is more evident.

4.3.2 Execution

After computing and comparing the RMSE values for each model, twe have chosen the best performing model for a detailed analysis of the model performance for each day of the week. By analyzing the average prediction error metric for each day of the week, we can determine in which situations the model underperforms.

Additionally, we have used several plots and figures to assess the method qualitatively. The basic evaluation measures we have introduced in 4.3.1 are not sufficient for a complete evaluation. We need to understand how the input features explain the workload prediction, how the error is distributed

Model	RMSE
Baseline	8.62
MLR	6.82
CNN	7.54

Table 4.1: RMSE of the workload prediction for each model

in relation to the identified peaks and how the predicted peaks relate to the observed peaks.

4.3.3 Results

After extending the model, we can fulfill the prediction task. After the modelling phase has been executed as outlined in Section 4.2.2, we have benchmarked the workload prediction models against the naive baseline with a RMSE of 8.62. With a RMSE of 6.82 for MLR predictions and a RMSE of 7.54 for the CNN, both predictions model perform better than the baseline for the overall prediction of workload (see Figure 4.1).

We have chosen the best performing model (MLR) to conduct a more robust analysis. For this purpose, we have calculated the RMSE separately for each day of the week, in order to assess in which conditions the model underperforms. We report the results in Table 4.2.

Day of the week	**RMSE**	**Average observed values**
Monday	7.40	15.42
Tuesday	7.91	14.15
Wednesday	8.96	18.8
Thursday	8.06	13.84
Friday	6.30	13.78
Saturday	3.22	2.26
Sunday	3.24	0

Table 4.2: RMSE of the MLR workload prediction for each day of the week

Visualizing the prediction outcomes against the observed values indicates inconsistent peak value prediction.

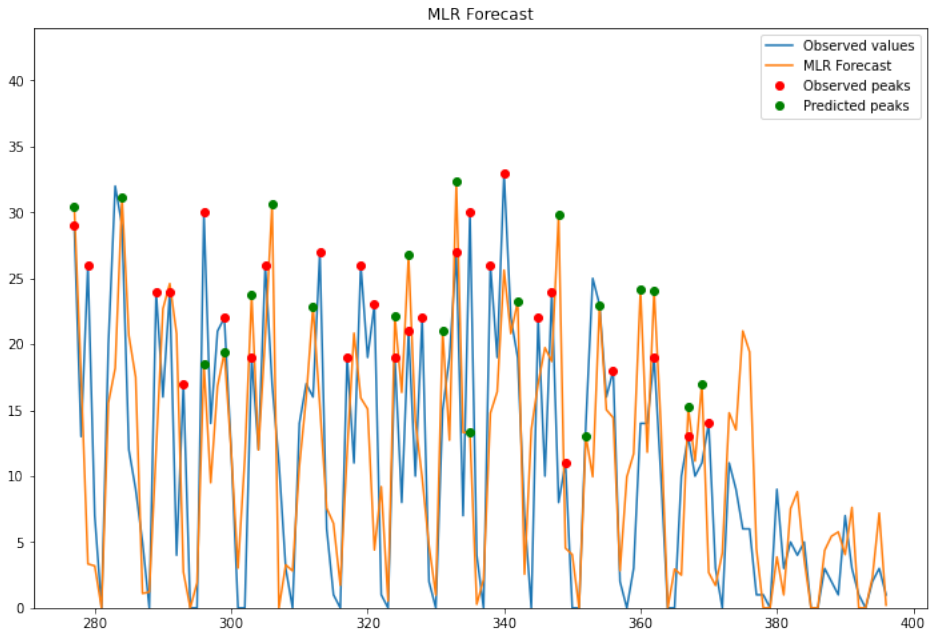


Figure 4.5: MLR workload predictions

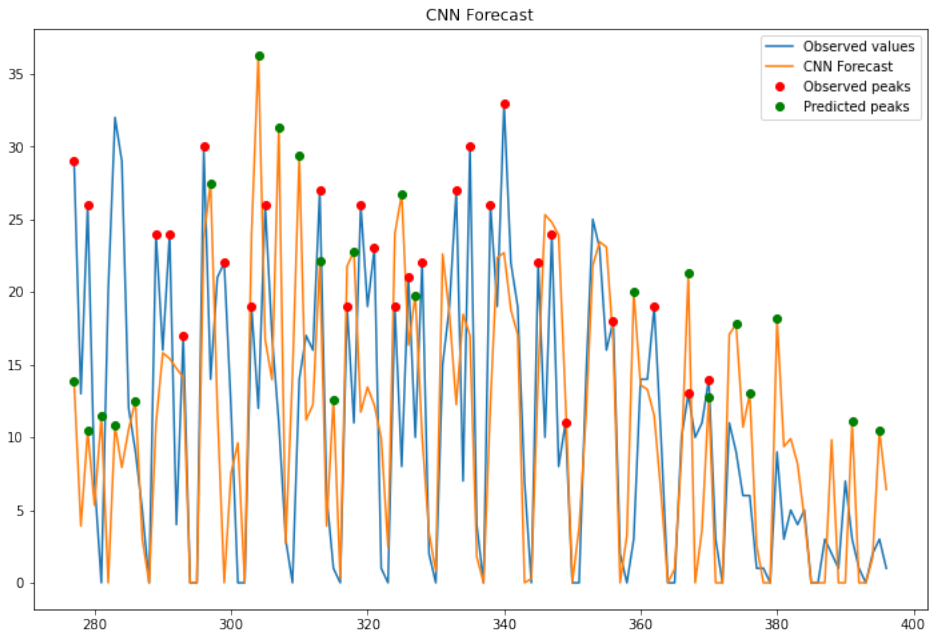


Figure 4.6: CNN workload prediction

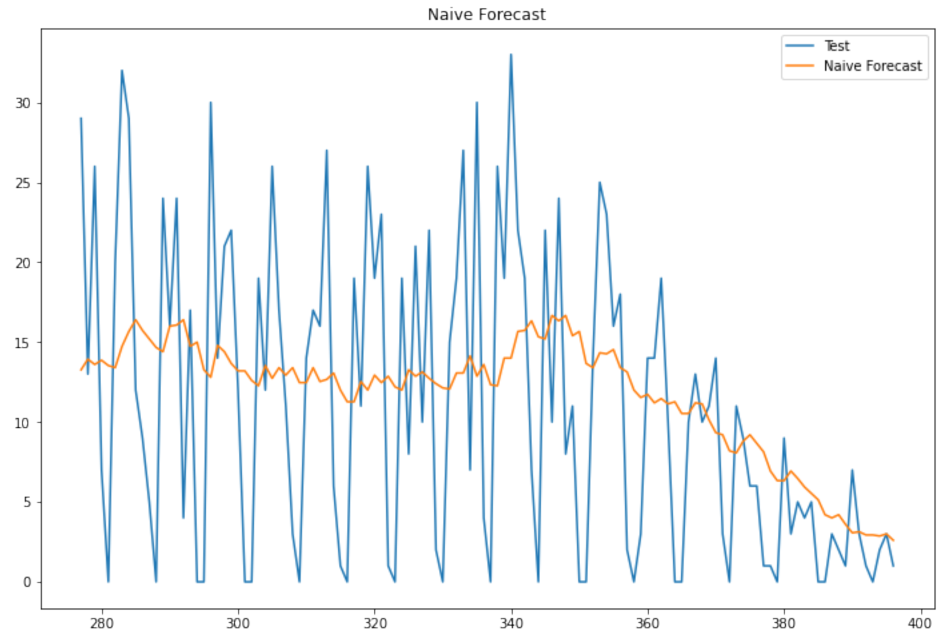


Figure 4.7: Naive workload prediction

Zooming in the analysis on the prediction of peaks, the results are summarized in the Confusion Matrices in Figures 4.8 and 4.9. We can calculate the Recall as shown in Section 4.3.2. With a recall of 41.37% for the peak detection corresponding to MLR predictions and a recall of 24.13% for the ones corresponding to the CNN predictions, the performance is below the pre-determined threshold in both cases.

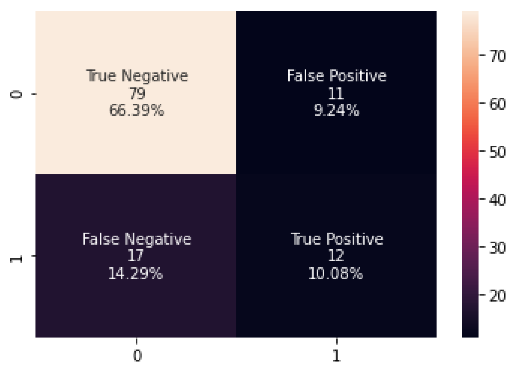


Figure 4.8: MLR true and predicted peaks

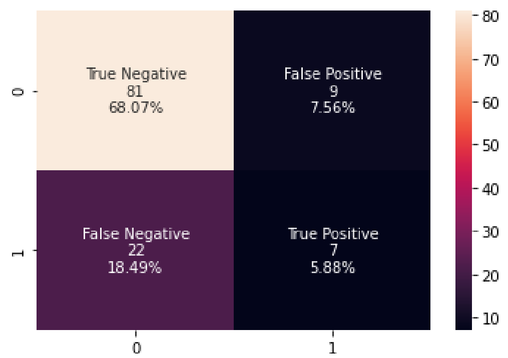


Figure 4.9: CNN true and predicted peaks

4.3.4 Discussion

In a broad sense, the PS-based workload prediction results seems to be satisfactory when compared to the naive baseline prediction output. This suggests that the features computed from the Historic Spectrum add value to the workload prediction. While comparatively the PS-based approach performs better than the naive baseline, the error measure is still large relative to the average workload (6.82 vs. 10.37). Furthermore, a more critical assessment of the results is required, as the correct prediction of extreme workload values is far more important than the prediction of inliers. Visualizing the predicted values against the observed values indicates that the peaks are predicted with similar frequency as the observed values. However, the peak prediction is inconsistent and unreliable. Even though the frequency is largely similar, in some cases, the peaks are predicted with 1-2 day delay or advance (see figures 4.5 and 4.6). The RMSE values for each day of the week, as seen in Table 4.2, suggest that the prediction of workload values during work week days is rather poor (see Table 4.2), which can be problematic given the increased business activity during the work week (as opposed to the weekend).

To investigate the performance of the models in more detail, we can conduct a qualitative assessment of the absolute error distribution. We plot the error distributions in figures 4.10 and 4.11. For both models we observe that peaks in workload prediction errors generally correspond to observed workload peaks or valleys. In other words, the models perform poorly in a volatile setting when we observe a sudden rise or fall in workload from one day to the next.

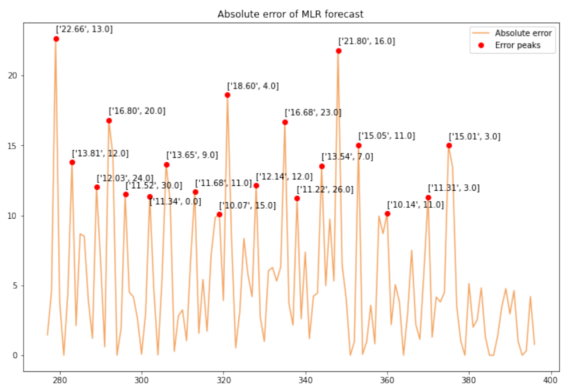


Figure 4.10: MLR error distribution. Error peaks are annotated with their value and the difference between the observed workload value at day x (x-coordinate of the peak) on the x-axis and the observed value at day x-1.

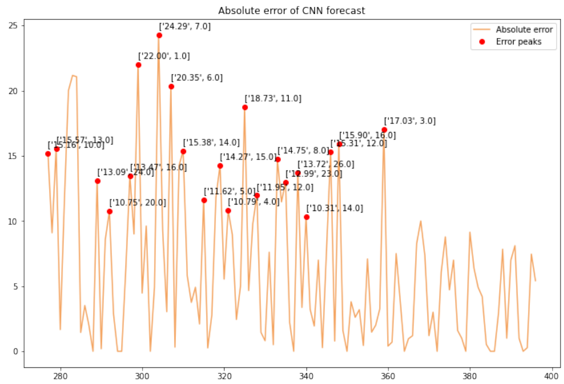


Figure 4.11: CNN error distribution. Error peaks are annotated with their value and the difference between the observed workload value at day x (x-coordinate of the peak) on the x-axis and the observed value at day x-1.

The recall metric clearly indicates a poor and unsatisfactory prediction of peak workload days. To understand the reason behind the poor performance, we have plotted each feature used for training next to the values of the dependent variable of the Target Spectrum. Given the correlations observed in figures 4.12 and 4.13 it can be concluded that the model underperformance could be caused by the poor selection of inter-case features. While the amount of workload of the Historic Spectrum seems to follow the same trend as the predicted workload, it is not sufficient for predicting extreme peaks. This leads to the conclusion that the expanded methodology is not suitable for business processes.

For the purpose of identifying points of improvement, we can visually assess the correlation between historic features and the target values by visualizing their distribution over time.

While largely, the general trend of increasing average workload towards the end of the year (followed by a sudden decrease during the 13th month) can be observed both in the case of historic aggregation of segments, as well as in the case of target aggregation of segments, it can be observed that the extreme target workload peaks cannot be explained by the historic workload. This can be seen in Figure 4.12. To investigate whether the other features are fit for predicting extreme values, we can inspect their distribution over time as well.

For reference, we define the following features:

- *maximum waiting time:* maximal delay between two consecutive W_Validate_Application activities in a day

$$max(t_{va,i+1} - t_{va,i}), \tag{4.3}$$

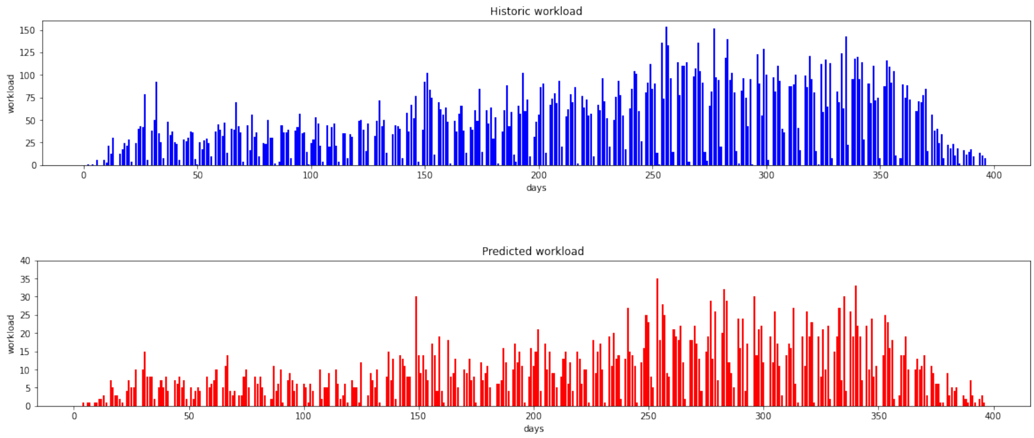


Figure 4.12: Distribution of historic and predicted workload over time

where va denotes W_Validate_ Application activities

- *workday length:* time between the first and last W_Validate_Application activities in a day
$$t_{va,last_d} - t_{va,first_d}, \tag{4.4}$$
where $last_{va,d}$ denotes the last W_Validate_Application activity of day d $first_{va,d}$ denotes the first W_Validate_Application activity of day d

- number of resources: number of distinct resources allocated to W_Validate_Application activities in a day

In figure 4.13 it can be observed that the resource perspective and time-related performance metrics (e.g. maximal delay between two W_Validate_Application activities in a day) seem to not exhibit similar trends (apart from day-of-the-week seasonality) to the ones observed in the distribution of the predictant. In fact, there seems to be no correlation between the extreme values of the predictors and the outliers of the response variable. This could explain the poor and inconsistent prediction of peak workload values.

In summary, we have determined that the original methodology is insufficient and inoperable on business processes. The extended methodology in turn satisfies the operability criteria, but it is not suitable for solving key prediction problems for businesses. The method performs poorly in the case of high volatility.

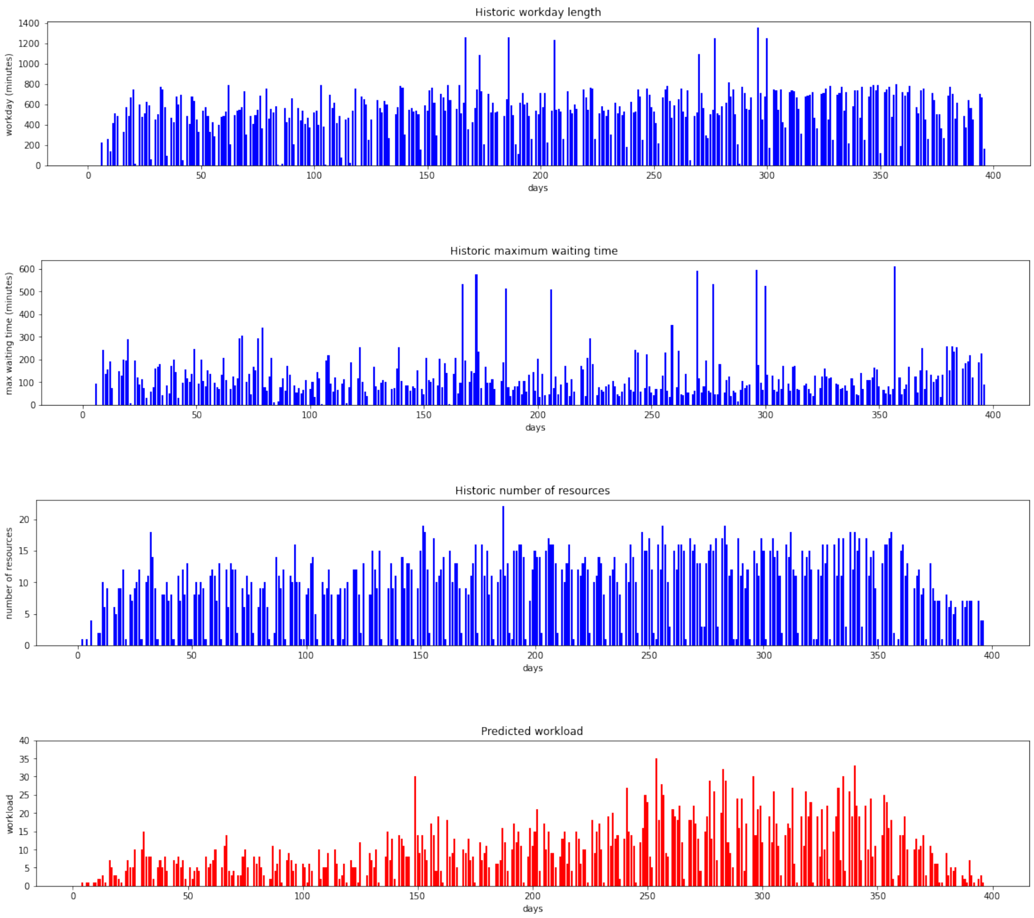


Figure 4.13: Distribution of several historic features (blue) and predicted workload over time (red)

Chapter 5

Conclusion

Following the evaluation discussion presented in Section 4.2.4, this research has achieved its goal of addressing the research question identified and outlined in 1.3. Specifically, it has been determined that the sequence in the approach of Denisov, Fahland and van der Aalst of existing steps can be maintained in their original sequence. However, we have identified the requirement of using formal process models along the feature selection process. Consequently, we have extended the methodology with 2 additional steps which take place at the beginning of the approach: business process understanding and event log pre-processing and filtering. After the addition of these steps, the method has become operable on business processes. Nevertheless, we have proved that the improvements are insufficient for the purpose of predicting sudden rises in values of workload.

The unsatisfactory prediction results point towards several improvement points. First, we have established and verified that process modelling plays a pivotal role in understanding of business processes. However, industry expertise is of utmost importance. As outlined in the PM^2 methodology [17], a proper PM project requires collaboration between business experts and process analysts. Further, the process analysis can be executed more thoroughly, including conformance checking and process enhancing. A suggested improvement point for inter-case feature selection is the selection of segments further upstream and downstream the target segments identified for aggregation, as such channels potentially hold trend-related information that could improve the prediction outcomes. Generally, we recommend a more careful approach to feature selection. Additionally, while the intra-case features are not essential for the operability of the model, they could potentially improve prediction of the PPI. The research has a proposed this as an optional step in the updated methodology. Further work could investigate the added value of intra-case features for aggregate business PPI prediction.

Bibliography

[1] What is a good credit score?, May 2021.

[2] Ana Isabel Rojão Lourenço Azevedo and Manuel Filipe Santos. Kdd, semma and crisp-dm: a parallel overview. *IADS-DM*, 2008.

[3] Melike Bozkaya, Joost Gabriels, and Jan Martijn van der Werf. Process diagnostics: a method based on process mining. In *2009 International Conference on Information, Process, and Knowledge Management*, pages 22–27. IEEE, 2009.

[4] V. Denisov, D. Fahland, and W. M. P. Van Der Aalst. Predictive performance monitoring of material handling systems using the performance spectrum. In *Proceedings - 2019 International Conference on Process Mining, ICPM 2019*, pages 137–144, 2019. Cited By :2.

[5] Vadim Denisov, Dirk Fahland, and Wil Aalst. Unbiased, fine-grained description of processes performance from event data: 16th international conference, bpm 2018, sydney, nsw, australia, september 9–14, 2018, proceedings. pages 139–157, 09 2018.

[6] Thomas G Dietterich. Machine learning for sequential data: A review. In *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*, pages 15–30, 2002.

[7] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A Reijers. Introduction to business process management. In *Fundamentals of business process management*, pages 1–31. Springer, 2013.

[8] Alfonso Marquez-Chamorro, Manuel Resinas, and Antonio Ruiz-Cortés. Predictive monitoring of business processes: A survey. *IEEE Transactions on Services Computing*, PP:1–1, 11 2017.

[9] Ricardo P Masini, Marcelo C Medeiros, and Eduardo F Mendes. Machine learning advances for time series forecasting. *arXiv preprint arXiv:2012.12802*, 2020.

[10] Elizaveta Povalyaeva, Ismail Khamitov, and A. Fomenko. Bpic 2017 : Density analysis of the interaction with clients. 2017.

[11] Arik Senderovich, Chiara Di Francescomarino, Chiara Ghidini, Kerwin Jorbina, and Fabrizio Maria Maggi. Intra and inter-case features in predictive process monitoring: A tale of two dimensions. In *International Conference on Business Process Management*, pages 306–323. Springer, 2017.

[12] Arik Senderovich, Chiara Di Francescomarino, and Fabrizio Maria Maggi. From knowledge-driven to data-driven inter-case feature encoding in predictive process monitoring. *Information Systems*, 84:255–264, 2019.

[13] Colin Shearer. The crisp-dm model: the new blueprint for data mining. *Journal of data warehousing*, 5(4):13–22, 2000.

[14] Wil Van Der Aalst. Data science in action. In *Process mining*, pages 3–23. Springer, 2016.

[15] Boudewijn van Dongen. Bpi challenge 2017, Feb 2017.

[16] de Medeiros A. K. A. Verbeek H. M. W. Weijters A. J. M. M. van der Aalst W. M. P. van Dongen, B. F. The prom framework: A new era in process mining tool support. In Gianfranco Ciardo and Philippe Darondeau, editors, *Applications and Theory of Petri Nets 2005*. Springer Berlin Heidelberg, 2005.

[17] Maikel L Van Eck, Xixi Lu, Sander JJ Leemans, and Wil MP Van Der Aalst. Pm

2

: a process mining project methodology. In *International Conference on Advanced Information Systems Engineering*, pages 297–313. Springer, 2015.

[18] T. van Meer. Pf4py. https://github.com/tomvmeer/PF4PY, 2020.

[19] Jan vom Brocke, Sarah Zelt, and Theresa Schmiedel. On the role of context in business process management. *International Journal of Information Management*, 36(3):486–495, 2016.

[20] Jolanta Žemgulienė and Mantas Valukonis. Structured literature review on business process performance analysis and evaluation. *Entrepreneurship and Sustainability Issues*, 6(1):226–252, 2018.

List of Figures

List of Tables

Appendix A

Example Event Log

'

	segment_name	start_day	case_id	end_day	duration
650	W_Call incomplete files:O_Returned	268	1783.0	268	0
651	W_Call incomplete files:O_Returned	268	1682.0	268	0
652	W_Call incomplete files:O_Returned	268	1772.0	268	0
653	W_Call incomplete files:O_Returned	268	1885.0	270	2
654	W_Call incomplete files:O_Returned	269	1830.0	269	0
655	W_Call incomplete files:O_Returned	269	1507.0	269	0
656	W_Call incomplete files:O_Returned	269	1838.0	269	0
657	W_Call incomplete files:O_Returned	269	1846.0	269	0
658	W_Call incomplete files:O_Returned	269	1765.0	269	0
659	W_Call incomplete files:O_Returned	269	1833.0	269	0
660	W_Call incomplete files:O_Returned	269	1749.0	269	0
661	W_Call incomplete files:O_Returned	269	1880.0	269	0
662	W_Call incomplete files:O_Returned	269	1774.0	270	1
663	W_Call incomplete files:O_Returned	270	1904.0	270	0
664	W_Call incomplete files:O_Returned	270	1869.0	271	1
665	W_Call incomplete files:O_Returned	270	1825.0	270	0
666	W_Call incomplete files:O_Returned	270	1879.0	270	0
667	W_Call incomplete files:O_Returned	270	1870.0	270	0
668	W_Call incomplete files:O_Returned	270	1834.0	270	0
669	W_Call incomplete files:O_Returned	270	1861.0	270	0
670	W_Call incomplete files:O_Returned	270	1437.0	270	0
671	W_Call incomplete files:O_Returned	270	1815.0	272	2
672	W_Call incomplete files:O_Returned	270	1860.0	270	0
673	W_Call incomplete files:O_Returned	270	1562.0	271	1
674	W_Call incomplete files:O_Returned	270	1866.0	270	0
675	W_Call incomplete files:O_Returned	270	1786.0	272	2
676	W_Call incomplete files:O_Returned	271	1770.0	277	6
677	W_Call incomplete files:O_Returned	271	1887.0	271	0
678	W_Call incomplete files:O_Returned	271	1923.0	275	4
679	W_Call incomplete files:O_Returned	271	1405.0	271	0

	segment_name	start_day	case_id	end_day	duration
680	W_Call incomplete files:O_Returned	271	1972.0	271	0
681	W_Call incomplete files:O_Returned	271	1938.0	276	5
682	W_Call incomplete files:O_Returned	271	1914.0	276	5
683	W_Call incomplete files:O_Returned	272	1824.0	272	0
684	W_Call incomplete files:O_Returned	272	1858.0	277	5
685	W_Call incomplete files:O_Returned	272	1592.0	272	0
686	W_Call incomplete files:O_Returned	273	1992.0	275	2
687	W_Call incomplete files:O_Returned	275	891.0	277	2
688	W_Call incomplete files:O_Returned	275	1986.0	276	1
689	W_Call incomplete files:O_Returned	275	1999.0	276	1
690	W_Call incomplete files:O_Returned	275	1530.0	277	2
691	W_Call incomplete files:O_Returned	275	1956.0	276	1
692	W_Call incomplete files:O_Returned	275	1895.0	275	0
693	W_Call incomplete files:O_Returned	276	1922.0	276	0
694	W_Call incomplete files:O_Returned	276	1886.0	276	0
695	W_Call incomplete files:O_Returned	276	1954.0	276	0
696	W_Call incomplete files:O_Returned	276	1863.0	276	0
697	W_Call incomplete files:O_Returned	276	1501.0	282	6
698	W_Call incomplete files:O_Returned	276	1998.0	276	0
699	W_Call incomplete files:O_Returned	276	1925.0	277	1
700	W_Call incomplete files:O_Returned	276	1855.0	276	0
701	W_Call incomplete files:O_Returned	276	1826.0	276	0
702	W_Call incomplete files:O_Returned	277	1901.0	277	0
703	W_Call incomplete files:O_Returned	277	1945.0	277	0
704	W_Call incomplete files:O_Returned	277	1856.0	277	0
705	W_Call incomplete files:O_Returned	277	2001.0	277	0
706	W_Call incomplete files:O_Returned	277	1571.0	277	0
707	W_Call incomplete files:O_Returned	277	1690.0	277	0
708	W_Call incomplete files:O_Returned	277	1997.0	277	0
709	W_Call incomplete files:O_Returned	277	1970.0	277	0
710	W_Call incomplete files:O_Returned	277	1989.0	277	0

	segment_name	start_day	case_id	end_day	duration
711	W_Call incomplete files:O_Returned	277	1948.0	278	1
712	W_Call incomplete files:O_Returned	277	1905.0	277	0
713	W_Call incomplete files:O_Returned	277	1910.0	278	1
714	W_Call incomplete files:O_Returned	277	1881.0	277	0
715	W_Call incomplete files:O_Returned	277	1976.0	277	0
716	W_Call incomplete files:O_Returned	277	1928.0	277	0
717	W_Call incomplete files:O_Returned	277	1773.0	277	0
718	W_Call incomplete files:O_Returned	277	1988.0	278	1
719	W_Call incomplete files:O_Returned	277	1963.0	277	0
720	W_Call incomplete files:O_Returned	277	1968.0	277	0
721	W_Call incomplete files:O_Returned	278	1969.0	278	0
722	W_Call incomplete files:O_Returned	278	1936.0	278	0
723	W_Call incomplete files:O_Returned	278	2010.0	278	0
724	W_Call incomplete files:O_Returned	278	1927.0	278	0
725	W_Call incomplete files:O_Returned	278	1962.0	278	0
726	W_Call incomplete files:O_Returned	278	1953.0	278	0
727	W_Call incomplete files:O_Returned	278	2018.0	278	0
728	W_Call incomplete files:O_Returned	278	1973.0	278	0
729	W_Call incomplete files:O_Returned	279	1964.0	279	0
730	W_Call incomplete files:O_Returned	279	1958.0	279	0
731	W_Call incomplete files:O_Returned	279	1921.0	279	0
732	W_Call incomplete files:O_Returned	279	1990.0	279	0
733	W_Call incomplete files:O_Returned	279	2004.0	279	0
734	W_Call incomplete files:O_Returned	279	1733.0	282	3
735	W_Call incomplete files:O_Returned	279	1727.0	279	0
736	W_Call incomplete files:O_Returned	279	2015.0	279	0
737	W_Call incomplete files:O_Returned	279	1993.0	279	0
738	W_Call incomplete files:O_Returned	279	1800.0	279	0
739	W_Call incomplete files:O_Returned	279	1844.0	282	3
740	W_Call incomplete files:O_Returned	280	2002.0	282	2
741	W_Call incomplete files:O_Returned	280	1892.0	284	4

	segment_name	start_day	case_id	end_day	duration
742	W_Call incomplete files:O_Returned	280	1514.0	283	3
743	W_Call incomplete files:O_Returned	280	1701.0	284	4
744	W_Call incomplete files:O_Returned	282	2021.0	282	0
745	W_Call incomplete files:O_Returned	282	1911.0	282	0
746	W_Call incomplete files:O_Returned	282	1995.0	282	0
747	W_Call incomplete files:O_Returned	282	1987.0	282	0
748	W_Call incomplete files:O_Returned	282	2023.0	283	1
749	W_Call incomplete files:O_Returned	282	1878.0	282	0

'

Appendix A: Example Event Log of
W_Call incomplete files:O_Returned segment

Appendix B

BPMN model of entire process

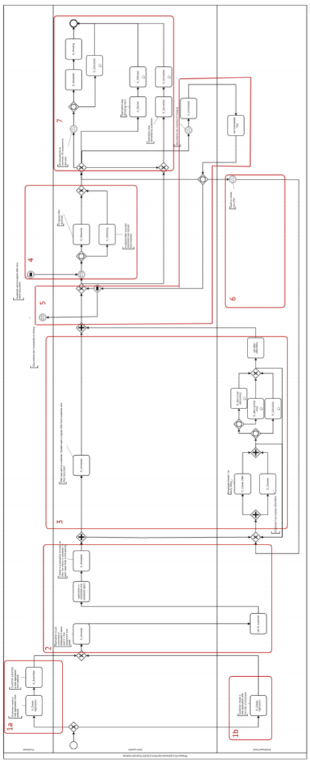


Figure B.1: BPMN model of the entire process, corresponding to the stages identified by Povalyaeva et al.

Appendix C

ProM Plugins

Plugin	**Settings**
Filter Event Log	1. Select traces by directly/ eventually follows event • Select attribute: concept:name • Select followed type: Directly followed • Select reference values: *W_Call_Incomplete_Files* • Select follower values: *O_Create_Offer* & *O_Returned* 2. Project traces on events by event frequency • Choose rate type: Frequency, 80% • Selection type: Filter in 3. Select traces by event attributes • Attribute type: concept:name • Selection type: Forbidden • Desired values: *A_Incomplete* & *O_Created*

Filter event log	4. Merge subsequent events • Select merge type: Merge taking first event • Select how to compare events: Compare event class • Select classifier: Event name • Select events: All, with the exception of workflow events (W prefix) and O_Create_Offer
Performance Spectrum Miner	• Bin size: 1d • Duration classifier: Quartile-based • Activity classifier: default
Interactive Data-aware heuristic miner (iDHM)	1. Process Model: C-net • Dependency Heuristic: Flexible Heuristic Miner • Conditional Heuristic: Cohen's Kappa • Bindings Heuristic: Nearest Activity • Conformance Heuristic: None • Frequency: 0.40 • Dependency: 0.9 • Bindings: 0.1 • Conditions: 0.5

InteractiveData-aware heuristic miner(iDHM)	2. Process Model: Directly-follows graph • Dependency Heuristic: Flexible Heuristic Miner • Conditional Heuristic: Cohen's Kappa • Bindings Heuristic: Nearest Activity • Conformance Heuristic: None • Frequency: 0.5 • Dependency: 0.9 • Bindings: 0.1 • Conditions: 0.5

Table C.1: ProM plugins used for process discovery and filtering of the event log